

BAB III

TEORI PENUNJANG

3.1. Status Gizi Balita

Menurut Almatsier (2003) Status gizi balita adalah keadaan keseimbangan antara asupan dan kebutuhan zat gizi. Status gizi baik terjadi ketika jumlah asupan zat gizi yang diterima sesuai dengan yang dibutuhkan oleh balita, dan begitu sebaliknya status gizi tidak seimbang dapat terjadi ketika asupan zat gizi yang diterima masih berkurang dari jumlah yang dibutuhkan. Sehingga status gizi merupakan batasan dalam memperkirakan keadaan tubuh dari jumlah penggunaan konsumsi makanan dan penggunaan zat-zat gizi.

Dalam menentukan status gizi balita harus ada ukuran baku yang digunakan yang sering disebut *reference*. Pengukuran baku antropometri yang sekarang digunakan di Indonesia adalah WHO-NCHS.

3.1.1. Macam Status Gizi Balita

Status gizi balita yang terdapat pada Desa Banjaragung Kecamatan Bareng Jombang berdasarkan hasil survei yaitu :

A. Gizi Baik (normal)

Status gizi baik atau status gizi *optimal* terjadi bila tubuh memperoleh cukup zat-zat gizi yang digunakan secara efisien sehingga memungkinkan pertumbuhan fisik, perkembangan otak, kemampuan kerja dan kesehatan secara umum pada tingkat setinggi mungkin (Almatsier, 2005). Gizi Seimbang terdapat pada susunan makanan yang dikonsumsi sehari-hari, mengandung zat-zat gizi yang berbeda-beda dan jumlah yang sesuai dengan kebutuhan tubuh, dengan memperhatikan prinsip keanekaragaman atau variasi makanan, aktivitas fisik yang dilakukan, kebersihan diri juga lingkungan, dan berat badan (BB). Bahan makanan yang dikonsumsi anak sejak usia dini merupakan fondasi penting bagi kesehatan dan kesejahteraan anak di masa depan.

Dengan kata lain, kualitas sumber daya manusia (SDM) hanya akan *optimal*, jika gizi dan kesehatan pada beberapa tahun kehidupannya di masa balita baik dan seimbang. Seperti pada gambar 3.1 kondisi balita yang mempunyai gizi baik dan sehat.



Gambar 3.1 balita gizi baik.

B. Gizi berkurang (gizi buruk)

Gizi buruk adalah suatu kondisi di mana seorang balita dinyatakan kekurangan nutrisi, atau dengan ungkapan lain status nutrisinya berada di bawah standar rata-rata. Nutrisi yang dimaksud bisa berupa protein, karbohidrat dan kalori. Jika anak mendapat asupan makanan yang cukup baik tetapi sering menderita diare atau demam, akhirnya jumlah zat gizi yang diterima selalu berkurang. Demikian juga pada anak yang makanannya tidak cukup (jumlah dan mutunya) maka daya tahan tubuhnya dapat melemah. Dalam keadaan demikian akan mudah diserang infeksi yang dapat mengurangi nafsu makan, akhirnya dapat menderita kurang gizi/ gizi buruk. seperti pada gambar 3.2 dibawah memperlihatkan balita yang memiliki gizi buruk (kurang baik) yang akan mengganggu tumbuh kembang balita pada umumnya.



Gambar 3.2 balita gizi buruk.

3.2. Data Mining

Data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Pramudiono, 2006). *Data mining* juga dapat diartikan sebagai aktifitas mengeksplorasi data dalam jumlah yang besar untuk menemukan pola dan aturan yang berarti. Salah satu Teknik yang dibuat dalam *data mining* adalah bagaimana menelusuri data yang ada untuk membangun sebuah model (Prasetyo, 2012)

Kata mining sendiri berarti usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar yang ada. Karena itu *Data Mining* sebenarnya memiliki akar yang Panjang dari bidang ilmu yang ada seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, *statistic* dan *database*. Semakin bertambahnya tahun atau bergantinya zaman *data mining* menjadi alat yang sangat penting untuk mengubah data tersebut menjadi informasi. Hal ini sering digunakan dalam berbagai praktek profil, seperti pemasaran, pengawasan, penemuan deteksi dan penemuan ilmiah, dan masih banyak lagi. *Data mining* juga telah digunakan selama berahun-tahun untuk bisnis oleh ilmuan, dan pemerintah untuk menyaring volume data seperti catatan perjalanan penumpang penerbangan pesawat, data sensus penduduk, dan supermarket dengan menggunakan scanner data untuk menghasilkan laporan riset pasar. Alasan utama untuk menggunakan *data mining* adalah untuk membantu mempermudah dalam analisis pengumpulan data dan koleksi pengamatan dalam berperilaku. Data tersebut rentan terhadap *collinearity* karena diketahui mempunyai keterkaitan. Fakta yang tak terelakkan *data mining* adalah bahwa subset / set data yang dianalisis mungkin tidak mewakili seluruh domain, dan karenanya tidak boleh berisi contoh-contoh hubungan kritis tertentu dan perilaku yang ada di bagian lain dari domain. Untuk mengatasi masalah semacam ini, analisis dapat ditambah menggunakan berbasis percobaan dan pendekatan lain, seperti *Choice Modelling* untuk data yang dihasilkan manusia. Dalam situasi ini, yang melekat dapat berupa korelasi kontrol atau data tidak dapat dihapus sama sekali, selama konstruksi desain eksperimental.

3.3. Pengelompokan Data Mining

Menurut (lasrore,2009), dalam pengelompokan *data mining* dapat dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu:

1. Diskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk mengembangkan pola dan kecenderungan yang sering memberikan kemungkinan dalam data, enkripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variable target estimasinya lebih kearah numeric dari pada kearah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variable prediksi.

3. Prediksi

Prediksi hamper sama dengan klasifikasi dan estimasi. Hanya saja terdapat perbedaan dalam prediksi nilai dari hasil aka nada dimasa mendatang.

4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori yang digunakan. Sebagai contoh, penggolongan pendapatan yang dapat dipisahkan dalam tiga kategori, yaitu : pendapatan tinggi, pendapatan sedang dan pendapatan rendah.

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengalaman, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain.

6. Asosiasi

Tugas asosiasi adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja yaitu menemukan barang yang akan dibeli secara bersamaan.

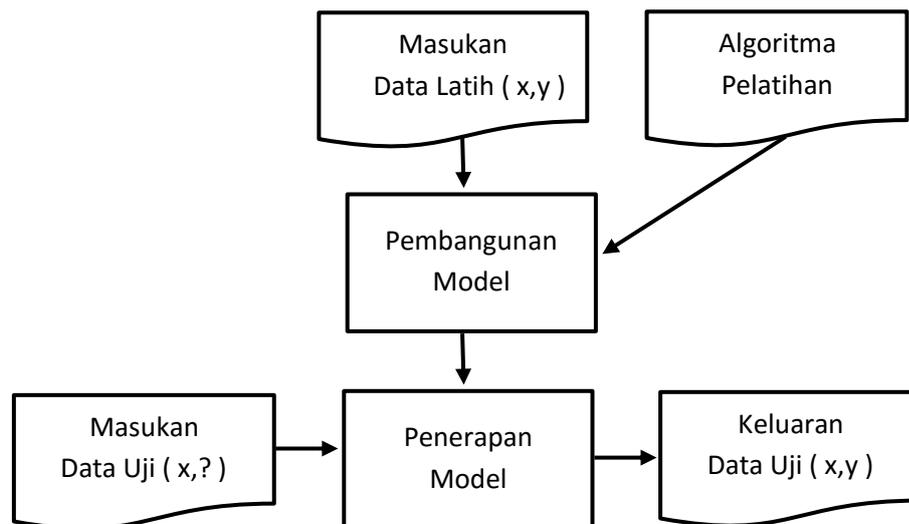
3.4. Klasifikasi

Klasifikasi merupakan pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dengan sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu pembangunan model sebagai protipe untuk disimpan sebagai memori dan penggunaan model tersebut untuk melakukan pengenalan / klasifikasi / prediksi pada suatu objek data lain agar diketahui dikelas mana objek data tersebut dalam model yang sudah disimpannya (Prasetyo, 2012).

Klasifikasi adalah metode *data mining* yang dapat digunakan untuk proses pencarian sekumpulan data model (fungsi) yang dapat menjelaskan dan membedakan kelas-kelas data atau konsep data, yang memiliki tujuan supaya model tersebut dapat digunakan untuk memprediksi objek kelas yang memiliki label yang nilainya tidak diketahui atau digunakan untuk memprediksi kecenderungan data-data yang sering muncul dimasa depan. Metode klasifikasi juga bertujuan untuk melakukan pemetaan data ke dalam kelas yang sudah didefinisikan sebelumnya berdasarkan pada nilai atribut data (Han dan Kember, 2006).

3.4.1. Model Dalam Klasifikasi

Model dalam klasifikasi didefinisikan secara detail sebagai suatu model pekerjaan yang prosesnya melakukan pelatihan / memerlukan model pembelajaran terhadap fungsi target, yang biasa diartikan sebagai suatu tempat untuk menerima masukan (data latih), kemudian mampu melakukan pemikiran terhadap masukan tersebut, dan memberikan jawaban sebagai keluaran dari hasil pemikirannya. Model tersebut dipakai untuk memprediksi kelas dari data uji. Proses pekerjaan dalam klasifikasi dapat dilihat di gambar 3.3.



Gambar 3.3 Proses pekerjaan klasifikasi (Prasetyo, 2012)

Dalam gambar 3.3 Model yang sudah dibangun pada saat melakukan masukan pelatihan kemudian dapat digunakan untuk memprediksi label kelas dari data baru yang belum diketahui label kelasnya. Dalam pembangunan model selama proses pelatihan tersebut diperlukan suatu algoritma untuk membangunnya, yang disebut algoritma pelatihan (*learning algorithm*). Ada beberapa algoritma pelatihan yang sudah dikembangkan oleh para peneliti antara lain *Naïve Bayes*, *Artificial Neuron Network*, *Support Vector Machine*, dan masih banyak lagi, kemudian penerapan model dengan mencoba memasukkan data uji kedalam model yang sudah dibuat dan keluaran data uji adalah hasil informasi yang dibutuhkan.

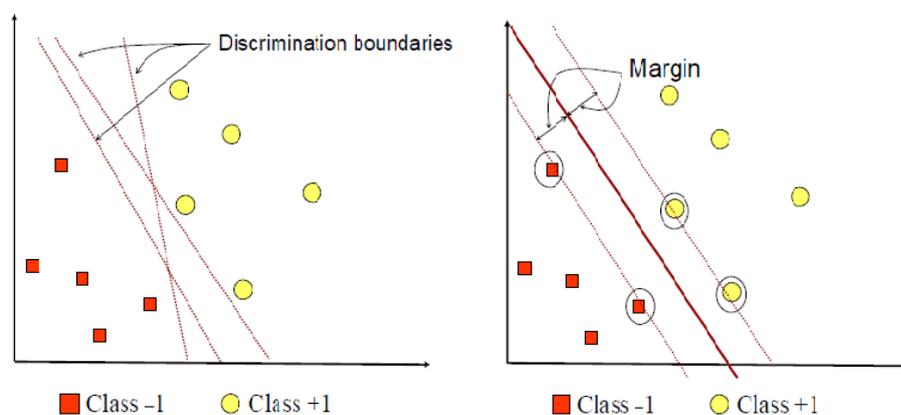
3.5. *Support Vector Machine*

Support Vector Machine (SVM) merupakan salah satu metode klasifikasi yang ada. SVM dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali di presentasikan pada tahun 1992 di Annual Workshop on Computational Learning Theory. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* (Duda dan Hart tahun 1973, Cover tahun 1965, Vapnik tahun 1964, dsb), *kernel* diperkenalkan oleh Aronszajn tahun 1950, dan demikian juga dengan konsep-konsep pendukung yang lain.

Akan tetapi hingga tahun 1992, belum pernah ada upaya merangkaikan komponen-komponen tersebut.

3.5.1. Konsep SVM

Konsep klasifikasi dengan SVM dapat dijelaskan secara sederhana sebagai suatu usaha untuk mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space*. *Pattern* yang merupakan anggota dari dua buah *class* : +1 dan -1 dan berbagai *alternative* garis pemisah (*discrimination boundaries*). *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Berbagai *alternative* garis pemisah ditunjukkan pada gambar *pattern* yang tergabung pada *class negative* disimbulkan dengan kotak, sedangkan *pattern* pada *class positive* disimbulkan dengan lingkaran. Proses pembelajaran dalam *problem* klasifikasi sebagai upaya menemukan garis yang memisahkan antara kedua kelompok tersebut. Berbagai *alternative* garis pemisah (*discrimination boundaries*) ditunjukkan pada gambar 3.4.



Gambar 3.4 garis pemisah dan Garis Hyperplane Terbaik

Hyperplane pemisah terbaik antara kedua *class* dapat dapat ditentukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. Pada gambar 3.4 bagian *margin* menunjukkan *hyperplane* terbaik yaitu terletak tepat ditengah-tengah kedua *class*. Upaya mencari lokasi *hyperplane optimal* merupakan inti dari proses pembelajaran SVM.

3.5.2. SVM Linear

Setiap data latih dinyatakan oleh (x_i, y_i) , dimana $i = 1, 2, \dots, N$, dan $x_i = \{x_{i1}, x_{i2}, \dots, x_{iq}\}$ merupakan atribut (fitur) set untuk data latih ke- i . $y_i \in \{-1, +1\}$ menyatakan label kelas. *Hyperplane* klasifikasi linier SVM, dinotasikan dengan :

$$w \cdot x_i + b = 0 \dots\dots\dots(3.1)$$

w dan b adalah parameter model. $w \cdot x_i$ merupakan inner-product dalam antara w dan x_i . data x_i yang masuk kedalam kelas -1 adalah data yang memenuhi pertidaksamaan berikut :

$$w \cdot x_i + b \leq -1 \dots\dots\dots(3.2)$$

Sementara data x_i yang masuk kedalam kelas + 1 adalah data yang memenuhi pertidaksamaan berikut :

$$w \cdot x_i + b \geq +1 \dots\dots\dots(3.3)$$

jika ada data dalam kelas -1 (misalnya x_a) yang bertempat di *hyperplane*, persamaan 3.1 akan terpenuhi. Untuk data kelas -1 dinotasikan dengan :

$$w \cdot x_a + b = 0 \dots\dots\dots(3.4)$$

sementara kelas +1 (misal x_b) akan memenuhi persamaan :

$$w \cdot x_b + b = 0 \dots\dots\dots(3.5)$$

dengan mengurangi persamaan 3.5 dengan 3.4 didapatkan :

$$w \cdot (x_b - x_a) = 0 \dots\dots\dots(3.6)$$

dengan memberikan label -1 untuk kelas pertama dan +1 untuk kelas kedua prediksi semua data uji akan menggunakan formula :

$$y = \begin{cases} +1 & \text{jika } w \cdot z + b > 0 \\ -1 & \text{jika } w \cdot z + b < 0 \end{cases} \dots\dots\dots(3.7)$$

untuk kelas -1 adalah data pada *support vector* yang memenuhi persamaan :

$$w \cdot x_a + b = -1 \dots\dots\dots(3.8)$$

untuk kelas +1 adalah data pada *support vector* yang memenuhi persamaan :

$$w \cdot x_b + b = 1 \dots\dots\dots(3.9)$$

dengan demikian *margin* dapat dihitung dengan mengurangi persamaan 3.8 dengan 3.9 didapatkan :

$$w \cdot (x_b - x_a) = 2 \dots\dots\dots(3.10)$$

margin hyperplane diberikan oleh jarak antara dua *hyperplane* dari dua kelas tersebut. Notasi diatas diringkas menjadi :

$$\|w\| \times d = 2 \text{ atau } \frac{2}{\|w\|} \dots\dots\dots(3.11)$$

3.5.3. Hyperplane SVM

Klasifikasi kelas data pada SVM pada persamaan 3.2 dan 3.3 dapat digabungkan dengan notasi :

$$y_i \cdot (w \cdot x_i - b) \geq 1, i = 1,2,\dots,N \dots\dots\dots(3.12)$$

margin optimal dihubungkan dengan memaksimalkan jarak antara *hyperplane* dan data terdekat. Jika jarak ini dirumuskan dengan persamaan 3.11 ($\|w\|$ adalah *Vector* bobot w). Selanjutnya diformulasikan ke dalam *quadratic programming (QP) problem*, dengan meminimalkan invers persamaan 3.12 $\frac{1}{2} \|w\|^2$, dibawah kendala (syarat) seperti berikut.

Minimalikan :

$$\frac{1}{2} \|w\|^2 \dots\dots\dots(3.13)$$

Syarat :

$$y_i \cdot (w \cdot x_i - b) \geq 1, i = 1,2,\dots,N \dots\dots\dots(3.14)$$

Optimalisasi ini dapat diselesaikan dengan *Lagrange Multiplier* :

$$L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i - b) - 1 \dots\dots\dots(3.15)$$

Keterangan :

N (banyaknya data).

α_i (*Lagrange Multiplier* yang berkorepondensi dengan x_i) harus diturunkan pada w , dan diset dengan nilai nol untuk syarat optimalisasi diatas.

L_p (*Lagrange Multiplier*).

Syarat 1 :

$$\frac{L_p = 0}{w} \rightarrow w = \sum_{i=1}^N \alpha_i y_i x_i \dots\dots\dots(3.16)$$

Syarat 2 :

$$\frac{L_p = 0}{b} \rightarrow w = \sum_{i=1}^N \alpha_i y_i = 0 \dots\dots\dots(3.17)$$

N adalah jumlah data yang menjadi *Support Vector Machine*. Karena Lagrange *Multiplier* (α) tidak diketahui nilainya, persamaan diatas tidak dapat diselesaikan secara langsung untuk mendapatkan w dan b. modifikasi dari 3.15 menjadi kasus pemaksimalan, syarat *optimal* menggunakan konstrain-Kuhn-Tucker (KKT) berikut :

Syarat 1 :

$$\alpha_i [y_i (w \cdot x_i + b)] = 0 \dots\dots\dots(3.18)$$

Syarat 2 :

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, N \dots\dots\dots(3.19)$$

Konstrain diatas menyatakan bahwa *Lagrange Multiplier* α_i nol, kecuali untuk data latih x_i yang memenuhi persamaan :

$$y_i (w \cdot x_i + b) = 1 \dots\dots\dots(3.20)$$

Dengan $\alpha_i > 0$ terletak pada *hyperplane* b_{i1} atau b_{i2} dan disebut *Support Vector*. Data latih yang tidak terletak di *hyperplane* tersebut mempunyai $\alpha_i = 0$. Persamaan *Lagrange Multiplier* 3.15 dijabarkan menjadi :

$$Lp = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i) - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i \dots\dots\dots(3.21)$$

Dengan mengganti nilai W maka Persamaan diatas akan menjadi dualitas *Lagrange Multiplier* berupa Ld dan didapatkan :

$$Ld = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \alpha_i \alpha_j y_i y_j x_i x_j \dots\dots\dots(3.22)$$

$x_i x_j$ merupakan *dot – product* dua data dalam data latih.

Syarat 1 :

$$\sum_{i=1}^N \alpha_i y_i = 0 \dots\dots\dots(3.23)$$

Syarat 2 :

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, N \dots\dots\dots(3.24)$$

Untuk set data yang besar, dapat diselesaikan dengan metode *Quadratic Programming* agar mendapatkan soludi yang layak untuk w dan b.

Hyperplane (batas keputusan) didapatkan dengan formula :

$$[\sum_{i=1}^N \alpha_i y_i x_i \cdot z] + b = 0 \dots\dots\dots(3.25)$$

N adalah jumlah data yang menjadi *Support Vector*. X_i merupakan *Support Vector*, z merupakan data uji yang akan diperdiksi kelasnya, dan $X_j \cdot z$ merupakan *inner-product* antara X_i dan z, karena α_i dihitung dengan metode

numerik dan mempunyai *error* numerik, nilai yang dihitung untuk b bisa jadi tidak sama. Biasanya diambil nilai rata-rata dari b (bias) yang didapat untuk menjadi parameter *hyperplane*. Untuk mendapatkan b , persamaan 3.18 dapat disederhanakan menjadi :

$$b_i = 1 - y_i (w \cdot x_i) \dots\dots\dots(3.26)$$

Pada umumnya kedua kelas tersebut tidak dapat terpisah secara sempurna, untuk itu persamaan dimodifikasi dengan memasukkan variabel ϵ_j , menjadi :

$$y_i \cdot (w \cdot x_i + b) \geq 1 - \epsilon_j. \dots\dots\dots (3.27)$$

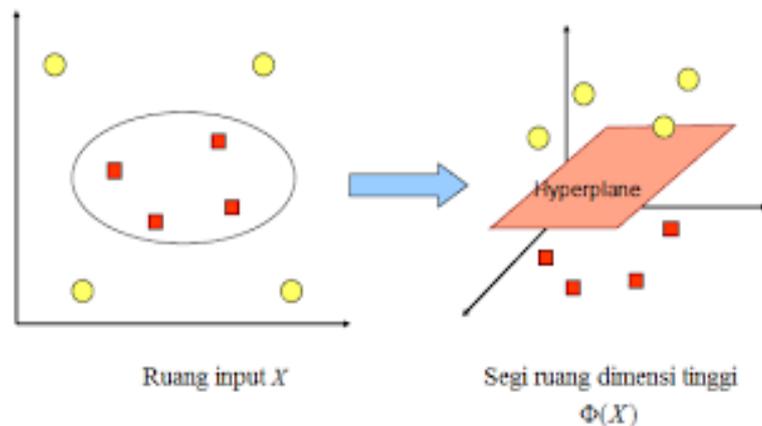
Demikian juga untuk masalah persamaan 3.13 :

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \epsilon_i. \dots\dots\dots (3.28)$$

Parameter C digunakan untuk mengontrol *trade-off* antara *margin* dan *error* klasifikasi. Semakin besar nilai C maka semakin besar pula pelanggaran yang dikenakan untuk tiap klasifikasi.

3.5.4 Kernel Trick dan Klasifikasi Non Linear pada SVM

Pada umumnya masalah dalam domain dunia nyata jarang yang berifat *linear*, kebanyakan bersifat *non linear*. Untuk menyelesaikan *problem non linear* SVM dimodifikasi dengan fungsi *kernel* dalam *non linear* SVM. Pertama-tama data x dipetakan oleh fungsi $\Phi(x)$ keruang *vector* yang berdimensi lebih tinggi. Pada ruang *vector* yang baru ini *hyperplane* yang memisahkan kedua *class* tersebut dapat dikonstruksikan, dimana kedua *class* dapat dipisahkan secara *linear* oleh sebuah *hyperplane*.



Gambar 3.5 fungsi Φ memetakan data keruang *vector* yang lebih tinggi

Pada gambar 3.5 pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada *input space* akan berjarak dekat juga pada *feature space*, sebaliknya dua data yang berjarak jauh pada *input space* akan juga berjarak jauh pada *feature space*. Sebagai suatu fungsi yang memetakan fitur data dari dimensi awal (rendah) ke fitur lain yang berdimensi lebih tinggi (bahkan jauh lebih tinggi). Φ merupakan fungsi *kernel* yang digunakan untuk pemetaan, D merupakan data latih, q merupakan set fitur dalam satu data yang lama, dan r merupakan set fitur yang baru sebagai hasil pemetaan untuk setiap data latih. Sementara x merupakan data latih, dimana $x_1, x_2, \dots, x_n \in D^r$ merupakan fitur yang akan dipetakan ke fitur berdimensi tinggi q , jadi untuk set data yang digunakan sebagai pelatihan dengan algoritma yang ada dari dimensi fitur yang lama r ke dimensi baru q . misalnya untuk *sampel* data N :

$$(\Phi(x_1), y_1, \Phi(x_2), y_2, \dots, \Phi(x_n), y_n) \in D^q \dots \dots \dots (3.29)$$

Pemetaan fitur lama pada set data ke fitur baru merupakan analogi dari hidden layer, dimana jumlah neuron dalam hidden layer biasanya mempunyai jumlah yang lebih banyak dibanding jumlah *vector* masukan.

Selanjutnya dilakukan proses pelatihan yang sama sebagaimana sebagai SVM *Linear*. Proses pemetaan pada fase ini memerlukan *dot-product* dua buah data pada ruang fitur baru. Nilai *dot-product* kedua buah *vector* ini dapat dihitung secara tidak langsung, yaitu tanpa mengetahui fungsi transformasi Φ . Teknik komputasi seperti ini disebut *kernel trick*, yaitu menghitung *dot-product* dua buah *vector* tersebut di ruang dimensi awal sebagai berikut :

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \dots \dots \dots (3.30)$$

Dan untuk prediksi pada set data dengan dimensi fitur yang baru diformulasikan :

$$F(\Phi(x)) = \text{sign}(w \cdot \Phi(z) + b) = \text{sign}(\alpha_1 y_1 \Phi(x_i) \cdot \Phi(z) + b) \dots \dots (3.31)$$

Kernel trick memberikan berbagai kemudahan, karena dalam proses pembelajaran SVM, untuk menentukan *support Vector*, kita hanya cukup mengetahui fungsi *kernel* yang dipakai, dan tidak perlu mengetahui wujud dari fungsi *non linear* Φ . Berbagai jenis fungsi *kernel* dikenal, sebagai mana dirangkumkan pada tabel 3.1

Tabel 3.1 Fungsi Kernel

Nama Kernel	Definisi Fungsi
Linear	$K(x,y) = x,y$
Polinomial	$K(x,y) = (x,y = c)^d$
Gaussian RBF	$K(x,y) = \exp \frac{\ x-y\ ^2}{2 \cdot \sigma^2}$
Sigmoid	$K(x,y) = \tanh(\sigma(x,y)+c)$
Invers Multikuadrik	$K(x,y) = \frac{1}{\sqrt{\ x-y\ ^2 + c^2}}$

Dijelaskan pada tabel 3.1, x dan y merupakan pasangan dua data dari semua bagian data latih. Parameter σ , c dan $d > 0$, merupakan konstanta $\|x-y\|^2$ merupakan kuadrat jarak antara *vector* X dan Y .

Pemilihan fungsi kernel yang tepat merupakan hal yang sangat penting, karena fungsi *karnel* akan menentukan fitur baru (dimensi tinggi) di mana fungsi klasifikasi *hyperplane* akan dicari.

3.5.5 Metode Sekuensial

Hyperplane optimal dalam SVM dapat ditemukan dengan merumuskannya *QP problem* dan diselesaikan dengan *library* yang tersedia dalam Analisa numerik. *Alternative* yang cukup sederhana dengan metode sekuensial yang telah dikembangkan oleh Vijayakumar, dengan penjelasan :

1. Inisialisasi $\alpha_i = 0$.

$$\text{Hitung matrix } D_{ij} = Y_i \cdot Y_j \cdot (K_{i,1} + \lambda^2) \dots \dots \dots (3.32)$$

$$(a) \delta \alpha_i = \min \{ \max [y (1 - E_i), -\alpha_i], C - \alpha_i \} \dots \dots \dots (3.33)$$

$$(b) \alpha_i = \alpha_i + \delta \alpha_i \dots \dots \dots (3.34)$$

2. lakukan step (a), (b), dan (c) untuk $i = 1, 2, \dots n$.

$$E_i = \sum_{j=1}^N \alpha_j D_{ij} \dots \dots \dots (3.35)$$

3. kembali ke step 2 sampai nilai α mencapai konvergen. Pada algoritma diatas, y adalah parameter untuk mengontrol kecepatan proses *learning*. Dan konvergen dapat didefinisikan dari tingkat perubahan nilai α .

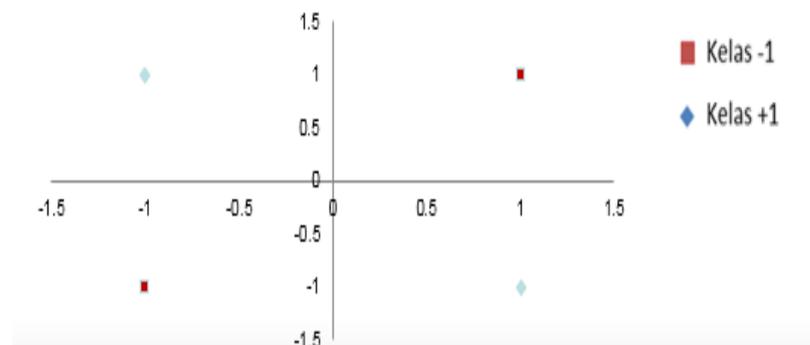
3.5.6 Contoh SVM Linear

Tentukan terlebih dahulu *Hyperplane* yang akan digunakan.

Tabel 3.2 SVM Linier

X_1	X_2	Kelas (y)	Support Vector
1	1	1	1
1	-1	-1	1
-1	1	-1	1
-1	-1	-1	0

Pada tabel 3.2 Pertama-tama jangkauan set data dan kelas yang akan dikonversi dulu agar sesuai dengan mensyaratkan bahwa kelas yang digunakan memakai nilai -1 dan +1. sehingga membentuk Visualisasi Data sebagai berikut :



Gambar 3.6 Bentuk Visualisasi Data

Dari data tabel 3.2 SVM Linier mendapatkan bentuk gambar diagram seperti pada gambar 3.6. Data akan dimasukkan pada diagram X_1 dan X_2 dengan nilai yang sudah ditentukan pada tabel 3.2 dengan syarat sebagai berikut :

- Karena ada dua fitur (x_1 dan x_2), maka w juga akan memiliki 2 fitur (w_1 dan w_2).
- Formulasi yang digunakan adalah sebagai berikut :

Meminimalkan nilai :

$$\frac{1}{2} \|w\|^2 = \frac{1}{2} (w_1^2 + w_2^2)$$

Dengan syarat :

$$y_i \cdot (w \cdot x_i - b) \geq 1, i = 1, 2, 3, \dots, N.$$

$$y_i \cdot (w_1 \cdot x_1 + w_2 \cdot x_2 + b) \geq 1.$$

Sehingga didapat persamaan sebagai berikut

$$1. (w_1 + w_2 + b) \geq 1, \text{ untuk } y_1 = 1, x_1 = 1, x_2 = 1.$$

$$2. (-w_1 + w_2 - b) \geq 1, \text{ untuk } y_2 = -1, x_1 = 1, x_2 = -1.$$

$$3. (w_1 - w_2 - b) \geq 1, \text{ untuk } y_3 = 1, x_1 = -1, x_2 = 1.$$

$$4. (w_1 + w_2 - b) \geq 1, \text{ untuk } y_4 = -1, x_1 = -1, x_2 = -1.$$

Dari persamaan pertama, kedua, ketiga dan keempat diatas, yaitu :

$$1. (w_1 + w_2 + b) \geq 1.$$

$$2. (-w_1 + w_2 - b) \geq 1.$$

$$3. (w_1 - w_2 - b) \geq 1.$$

$$4. (w_1 + w_2 - b) \geq 1.$$

Dilanjutkan persamaan satu dan dua sebagai berikut :

$$(w_1 + w_2 + b) \geq 1$$

$$(-w_1 + w_2 - b) \geq 1 \quad +$$

$$\hline 2w_2 = 2$$

$$w_2 = 2 / 2$$

$$w_2 = 1$$

$$\text{Maka } w_2 = 1$$

selanjutnya menyederhanakan persamaan satu dan tiga :

$$(w_1 + w_2 + b) \geq 1$$

$$(w_1 - w_2 - b) \geq 1 \quad +$$

$$\hline 2w_1 = 2$$

$$w_1 = 2 / 2$$

$$w_1 = 1$$

$$\text{Maka } w_1 = 1$$

Menyederhanakan Persamaan (2) dan (3)

$$(-w_1 + w_2 - b) \geq 1$$

$$(w_1 - w_2 - b) \geq 1 \quad +$$

$$\hline -2b = 2$$

$$b = 2 / -2$$

$$b = -1$$

$$\text{Maka } b = -1$$

Sehingga didapatkan persamaan *hyperplane* :

$$w_1 x_1 + w_2 x_2 + b = 0$$

$$x_1 + x_2 - 1 = 0$$

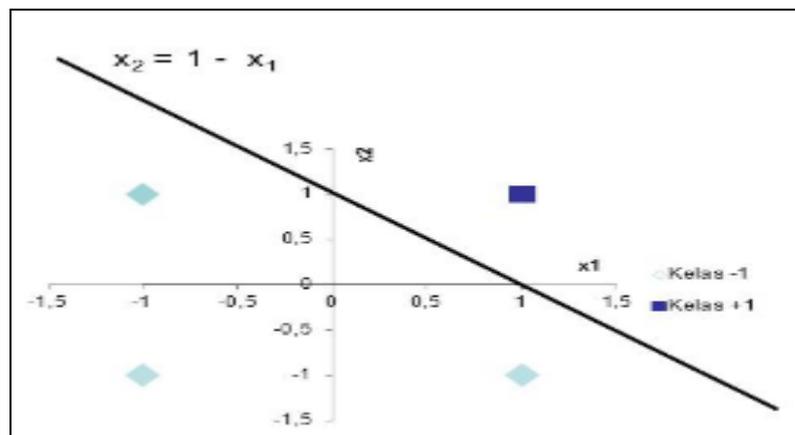
$$x_2 = 1 - x_1$$

Nilai persamaan $x_2 = 1 - x_1$ akan dihitung lagi dengan mengganti nilai x_1 seperti pada tabel 3.4 *Hyperplane*. Visualisasi garis *Hyperplane* (sebagai Fungsi Klasifikasi) :

Tabel 3.3 *Hyperplane*

X_1	$X_2 = 1 - X_1$
-2	3
-1	2
0	1
1	0
2	-1

Pada tabel 3.4 dari hasil persamaan selanjutnya menghitung nilai X_2 dengan nilai X_1 yang sudah misalkan seperti pada tabel 3.4. Jika dari persamaan $X_2 = 1 - X_1$ dimisalkan nilai $X_1 = 1$ maka dapat ditarik garis *hyperplane* seperti gambar 3.6 sebagai berikut :



Gambar 3.7 Visualisasi garis *Hyperplane*

Gambar 3.6 adalah contoh hasil visualisasi garis *hyperplane* yang didapat dari tabel dengan persamaan $X_2 = 1 - X_1$. visualisasi garis *hyperplane* X_2 yang didapat akan seperti gambar dengan menarik garis melalui nilai yang sudah dihasilkan.