

BAB II

LANDASAN TEORI

2.1. Pembangunan Manusia

Manusia pada dasarnya adalah kekayaan yang dimiliki suatu negara. Tujuan dari sebuah pembangunan adalah menciptakan lingkungan yang memungkinkan bagi rakyatnya untuk menikmati umur panjang, sehat, dan menjalankan kehidupan produktif. Tetapi hal yang seringkali terlupakan dalam diri manusia adalah kesibukan dalam memperkaya diri, mengumpulkan harta dan uang, sehingga kurang dalam mengembangkan diri dan menambah pengetahuan.

Dalam UNDP (*United Nations Development Programme*), pembangunan manusia adalah suatu proses untuk memperbesar pilihan-pilihan bagi manusia. Pada saat yang sama pembangunan manusia dapat dilihat juga sebagai pembangunan (*formation*) kemampuan manusia melalui perbaikan taraf kesehatan, pengetahuan, dan ketrampilan, sekaligus pemanfaatan kemampuan/ketrampilan mereka.

Menurut UNDP dalam *Human Development Report* (HDR) yang menekankan bahwa untuk memperluas pilihan-pilihan manusia, konsep pembangunan manusia harus dibangun dari empat dimensi yang tidak terpisahkan. Berdasarkan konsep di atas maka untuk menjamin tercapainya tujuan pembangunan manusia, ada empat unsur pokok yang perlu diperhatikan (UNDP: 1995) yaitu:

a. Produktivitas (*Productivity*)

Masyarakat harus mampu untuk meningkatkan produktifitas mereka dan berpartisipasi penuh dalam proses mencari penghasilan dan lapangan pekerjaan. Oleh karena itu, pembangunan ekonomi merupakan bagian dari model pembangunan manusia.

b. Pemerataan (*equity*)

Masyarakat harus mempunyai akses untuk memperoleh kesempatan yang adil. Semua hambatan terhadap peluang ekonomi dan politik harus dihapuskan sehingga masyarakat dapat berpartisipasi di dalam dan memperoleh manfaat dari peluang-peluang yang ada.

c. Kestinambungan (*Sustainability*)

Akses untuk memperoleh kesempatan harus dipastikan bahwa tidak hanya untuk generasi sekarang tetapi juga untuk generasi yang akan datang. Semua jenis pemodalan baik itu fisik, manusia, dan lingkungan hidup harus dilengkapi.

d. Pemberdayaan (*Empowerment*)

Pembangunan harus dilakukan oleh masyarakat, dan bukan hanya untuk mereka. Masyarakat harus berpartisipasi penuh dalam mengambil keputusan dan prosesproses yang memengaruhi kehidupan mereka.

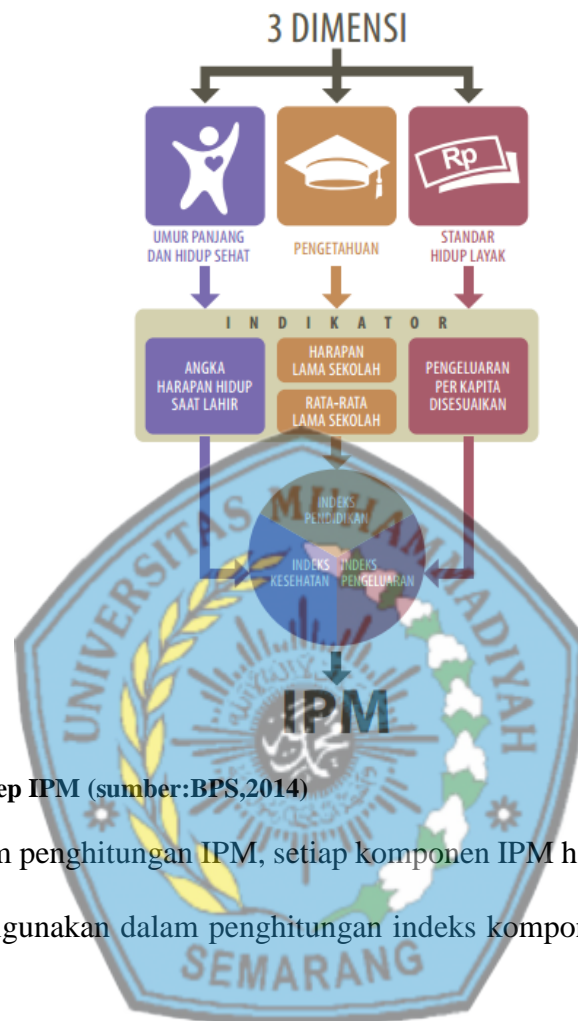
Berdasarkan pengertian-pengertian di atas maka dapat disimpulkan bahwa pelayanan merupakan suatu bentuk sistem, *procedure* atau metode tertentu diberikan kepada orang lain. Dalam hal ini, kebutuhan pelanggan tersebut dapat terpenuhi sesuai dengan harapan atau keinginan pelanggan dengan tingkat persepsi mereka.

2.2. Indeks Pembangunan Manusia (IPM)

Menurut Badan Pusat Statistik (2008) Indeks Pembangunan Manusia (IPM) adalah mengukur capaian pembangunan manusia manusia berbasis sejumlah komponen dasar kualitas hidup. Indeks pembangunan manusia mengukur capaian pembangunan manusia berbasis sejumlah komponen dasar kualitas hidup. Sebagai ukuran kualitas hidup, IPM dibangun melalui tiga dimensi dasar yaitu:

- a. Dimensi umur panjang dan hidup sehat.
- b. Dimensi pengetahuan.
- c. Dimensi standar hidup layak (BPS: 2014)

Untuk mengukur dimensi umur panjang dan hidup sehat digunakan angka harapan hidup waktu lahir. Selanjutnya untuk mengukur dimensi pengetahuan digunakan gabungan indikator harapan lama sekolah dan rata-rata lama sekolah. Adapun untuk mengukur dimensi hidup layak digunakan indikator pengeluaran perkapita disesuaikan sebagai pendekatan pendapatan yang mewakili capaian pembangunan untuk hidup layak.



Gambar 2. 1 Konsep IPM (sumber:BPS,2014)

Sebelum penghitungan IPM, setiap komponen IPM harus dihitung indeksnya. Formula yang digunakan dalam penghitungan indeks komponen IPM adalah sebagai berikut:

$$\text{Dimensi Kesehatan: } I_{\text{Kesehatan}} = \frac{AHH - AHH_{\min}}{AHH_{\max} - AHH_{\min}} \quad (1)$$

$$\text{Dimensi Pendidikan: } I_{\text{HLS}} = \frac{HLS - HLS_{\min}}{HLS_{\max} - HLS_{\min}} \quad (2)$$

$$I_{\text{RLS}} = \frac{RLS - RLS_{\min}}{RLS_{\max} - RLS_{\min}} \quad (3)$$

$$\text{Dimensi Pengeluaran } I_{\text{pengeluaran}} = \frac{\ln(\text{pengeluaran}) - \ln(\text{pengeluaran}_{\min})}{\ln(\text{pengeluaran}_{\max}) - \ln(\text{pengeluaran}_{\min})} \quad (4)$$

Untuk menghitung indeks masing-masing komponen IPM digunakan batas maksimum dan minimum seperti terlihat dalam Tabel berikut:

Tabel 2. 1 Batas Maksimum dan Minimum(Sumber: BPS, 2014)

No	Komponen IPM	Maksimum	Minimum	Keterangan
1	Angka Harapan Hidup Saat Lahir(Tahun)	85	20	Standar UNDP
2	Angka harapan Lama Sekolah(Tahun)	18	0	Standar UNDP
3	Rata-rata Lama Sekolah (Tahun)	15	0	Standar UNDP
4	Pengeluaran Disesuaikan Perkapita	26.572.352**(Rp)	1.007.436*(Rp)	Pengeluaran per Kapita Disesuaikan

Keterangan:

- * Daya beli minimum merupakan garis kemiskinan terendah kabupaten tahun 2010(data empiris yaitu di Tolikara-Papua)
- ** Daya beli maksimum merupakan nilai tertinggi kabupaten yang diproyeksikan hingga 2025 (akhir RPJPN) yaitu perkiraan pengeluaran per kapita Jakarta Selatan tahun 2025 (Sumber: BPS: 2014)

Selanjutnya nilai IPM dapat dihitung sebagai:

$$IPM = \sqrt[3]{I_{kesehatan} \times I_{pendidikan} \times I_{pengeluaran}} \times 100 \quad (5)$$

Variabel yang mempengaruhi indeks pembangunan manusia adalah:

2.2.1. Indeks Harapan Hidup

Angka Harapan Hidup (AHH) merupakan alat untuk mengevaluasi kinerja pemerintah dalam meningkatkan kesejahteraan penduduk pada umumnya, dan meningkatkan derajat kesehatan pada khususnya (BPS: 2014). Angka Harapan Hidup

yang rendah di suatu daerah harus diikuti dengan program pembangunan kesehatan, dan program sosial lainnya termasuk kesehatan lingkungan, kecukupan gisi dan kalori termasuk program pemberantasan kemiskinan. Angka harapan hidup mencerminkan drajat kesehatan suatu masyarakat.

2.2.2. Indeks Pendidikan

Salah satu komponen pembentuk IPM adalah dari dimensi pengetahuan yang diukur melalui tingkat pendidikan. Dalam hal ini, indikator yang digunakan adalah rata-rata lama sekolah (*mean years of schooling*) dan angka harapan lama sekolah. Rata-rata lama sekolah didefinisikan sebagai jumlah yang digunakan oleh penduduk dalam menjalani pendidikan formal (BPS: 2014). Diasumsikan bahwa dalam kondisi normal rata-rata lama sekolah suatu wilayah tidak akan turun. Cakupan penduduk yang dihitung dalam perhitungan rata-rata lama sekolah adalah penduduk berusia 25 tahun ke atas.

Angka harapan lama sekolah didefinisikan lamanya sekolah (dalam tahun) yang diharapkan akan dirasakan oleh anak pada umur tertentu di masa mendatang (BPS: 2014). Diasumsikan bahwa peluang anak tersebut akan tetap bersekolah pada umur-umur berikutnya sama dengan peluang penduduk yang bersekolah per jumlah penduduk untuk umur yang sama saat ini.

Angka harapan lama sekolah dihitung untuk penduduk berusia 7 tahun ke atas. Harapan lama sekolah dapat digunakan untuk mengetahui kondisi pembangunan sistem pendidikan di berbagai jenjang yang ditunjukkan dalam bentuk lamanya pendidikan (dalam tahun) yang diharapkan dapat dicapai oleh setiap anak.

2.2.3. Indeks Standar Hidup Layak

Dimensi lain dari ukuran kualitas hidup manusia adalah standar hidup layak. Dalam cakupan lebih luas, standar hidup layak menggambarkan tingkat kesejahteraan yang dinikmati oleh penduduk sebagai dampak semakin membaiknya ekonomi (BPS: 2014). UNDP mengukur standar hidup layak menggunakan Produk Domestik Bruto (PDRB) riil yang disesuaikan, sedangkan BPS dalam menghitung standar hidup layak menggunakan Pengeluaran rata-rata disesuaikan.

Pengeluaran per kapita yang disesuaikan ditentukan dari nilai pengeluaran per kapita dan paritas daya beli (*Purchasing Power Parity*). Rata-rata pengeluaran per kapita setahun diperoleh dari Susenas, dihitung dari level provinsi hingga level kab/kota. Rata-rata pengeluaran per kapita dibuat konstan/riil dengan tahun dasar 2012=100. Perhitungan paritas daya beli pada metode baru menggunakan 96 komoditas dimana 66 komoditas merupakan makanan dan sisanya merupakan komoditas nonmakanan. Metode penghitungan paritas daya beli menggunakan Metode Rao. Perhitungan paritas daya beli dihitung dari bundle komoditas makanan dan non makanan.

Rumus Perhitungan Paritas Daya Beli (PPP):

$$PPP_j = \prod_{i=1}^m \left(\frac{p_{ij}}{p_{ik}} \right)^{\frac{1}{m}} \quad (6)$$

Keterangan

P_{ij} = Harga komoditas i di kab atau kota j

P_{ik} = Harga komoditas i di Jakarta Selatan

m = Jumlah komoditas

Pengelompokan indeks pembangunan manusia (BPS, 2014):

IPM < 60 : IPM rendah

$60 \leq$ IPM < 70 : IPM sedang

$70 \leq$ IPM < 80 : IPM tinggi

IPM \geq 80 : IPM sangat tinggi

2.3. Data Mining

Data mining adalah suatu teknologi terbaru dimana menggabungkan metode analisis tradisional dengan algoritma yang mampu memroses data dengan jumlah besar. Data mining adalah suatu istilah yang digunakan untuk mencari informasi yang tersembunyi didalam data yang besar. Data mining merupakan metode yang menggabungkan tiga ilmu yaitu statistik, matematika, kecerdasan buatan, dan machine learning. Beberapa definisi data mining antara lain, data mining disebut juga serangkaian proses untuk menggali suatu nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data yang sangat besar (Prambudiono, 2007). Data mining atau *machine learning* adalah strategi semi empiris yang menggunakan data tentang sifat/*properties* dan deskriptor (Lu *et al.*, 2017).

Analisis yang dilakukan data mining akan melebihi cepat untuk mengambil keputusan. Data mining mengeksplor data untuk menemukan pola-pola tersembunyi, mencari informasi untuk memperderiksi yang akan terjadi. Data mining memiliki bebrapa sebutan diantaranya: *Knowledge discovery* (mining) in databases (KDD), ekstraksi pengetahuan, analisis data/pola, dan kecerdasan bisnis.

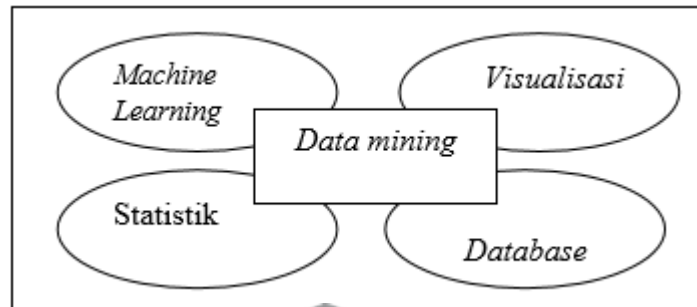
Karakteristik yang terkandung dalam data mining adalah:

- a. Data mining berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang diketahui sebelumnya.
- b. Data mining biasa menggunakan data yang berjumlah sangat besar. Data yang besar akan dapat dipercaya kesimpulannya.
- c. Data mining sangat berguna untuk membuat suatu keputusan yang kritis, terutama strategi (Davies, 2004).

Dalam data mining terdapat istilah pola dimana arti dari sebuah pola adalah suatu ilmu yang mempelajari pengklasifikasian suatu objek dalam beberapa kelas atau kategori dan mengenali kecendrungan data. Tergantung pda penerapannya, objek-objek ini berupa pasien, mahasiswa, permohonan kredit, *image* atau *signal* atau pengukuran lain yang perlu diklasifikasikan dan dicari fungsi regresinya (Santoso, 2007).

Machine Learning adalah suatu daerah atau area dalam *artificial intelligence* atau kecerdasan buatan yang saling terkait dengan teknik-teknik yang bisa ditransformasikan kedalam bentuk program. *Machine learning* adalah alat untuk analisis dalam data mining. Berikut gambaran posisi data mining yang merupakan

gabungan dari *machine learning*, *visualisasi*, *Statistik*, *Database* dapat dilihat pada gambar 2.2 (Santoso, 2007).



Gambar 2. 2 Posisi Data Mining dalam Ilmu Statistik

Pekerjaan yang berkaitan erat dengan data mining adalah model prediksi (*prediction modelling*), analisis kelompok (*cluster analysis*), analisis asosiasi (*association analysis*), dan deteksi anomaly (Prasetyo, 2012).

a. Model Prediksi

Model prediksi berkaitan erat dengan pembuatan suatu model yang dapat membentuk pemetaan dari suatu himpunan variabel ke setiap targetnya, kemudian menggunakan model tersebut untuk memberikan suatu nilai pada himpunan baru yang didapat. Dalam model prediksi terdapat dua jenis yaitu klasifikasi dan regresi. Model klasifikasi digunakan untuk variabel target diskrit, sedangkan model regresi variabel kontinu.

b. Analisis kelompok

Analisis kelompok mengelompokkan data kedalam sejumlah kelompok berdasarkan karakteristik masing-masing data pada kelompok data yang

sama. Data-data yang masuk dalam batasan-batasan kelompok akan menggrombol dan terpisah dengan kelompok yang berbeda karakteristiknya.

c. Analisis Asosiasi

Analisis asosiasi berguna untuk menemukan pola yang menggambarkan suatu hubungan fitur dalam data. Setelah pola ditemukan biasanya merepresentasikan bentuk subset fitur.

d. Deteksi Anomali

Fungsi dari deteksi anomaly adalah mendeteksi data yang berbeda dengan data yang lainnya atau dengan kata lain data *outlier*.

2.4. Support Vector Machine (SVM)

Support vector machine (SVM) adalah suatu teknik yang relatif baru untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM adalah solusi global optimal dan menghindari *deminsionalitas* (Wu and FAN, 2013). SVM berada dalam satu kelas dengan ANN dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *supervised learning*, dimana dalam implementasinya perlu adanya tahap *training* dan disusul tahap *testing*. Implementasi SVM berasal dari prinsip *structural risk minimization* (Yuan, 2015).

Baik para ilmuwan maupun praktisi telah banyak menerapkan teknik ini dalam menyelesaikan masalah-masalah nyata dalam kehidupan sehari-hari. Penerapannya antara lain dalam masalah *gene expression analysis*, prediksi finansial, cuaca hingga bidang kedokteran. Terbukti dalam banyak implementasi, SVM memberi hasil yang

lebih baik dari ANN, terutama dalam hal solusi yang dicapai. ANN menemukan solusi berupa *local optimal*, sedangkan SVM menemukan solusi yang *global optimal*. Jika menggunakan ANN, solusi dari setiap training hampir selalu berbeda. Hal ini disebabkan solusi *local optimal* yang dicapai tidak selalu sama. SVM selalu mencari solusi yang sama untuk setiap running. Teknik berusaha untuk menemukan fungsi pemisah (klasifier) yang *optimal* yang bisa memisahkan dua set data dari dua kelas yang berbeda (Vapnik, 2000). Berikut fungsi linier dari SVM:

$$g(x) := \text{sgn}(f(x)) \quad (7)$$

Dengan

$$f(x) = w^T x + b$$

$x, w \in R^n$ dan $b \in R$. Masalah klasifikasi ini bisa dirumuskan sebagai berikut: untuk menemukan nilai dari parameter (w, b) maka $\text{sgn}(f(x_i)) = \text{sgn}(\langle w, x \rangle + b) = y_i$ untuk semua i . Fungsi sgn digunakan untuk mengelompokkan semua nilai di atas 0 menjadi +1 dan dibawah 0 menjadi -1. Metode SVM berusaha menemukan fungsi pemisah (klasifier/*hyperplane*) terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam obyek. *Hyperplane* terbaik adalah *hyperplane* yang terletak di tengah-tengah antara dua set obyek dari dua kelas. Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan margin atau jarak antara dua set obyek dari kelas yang berbeda. Jika $w x_1 + b = +1$ adalah *hyperplane*-pendukung (*supporting hyperplane*) dari kelas +1, dan $w x_2 + b = -1$ adalah *hyperplane*-pendukung dari kelas -1, margin antara dua kelas dapat dihitung

dengan mencari jarak antara kedua hyperplane-pendukung dari kedua kelas. Secara spesifik, margin dihitung dengan cara berikut:

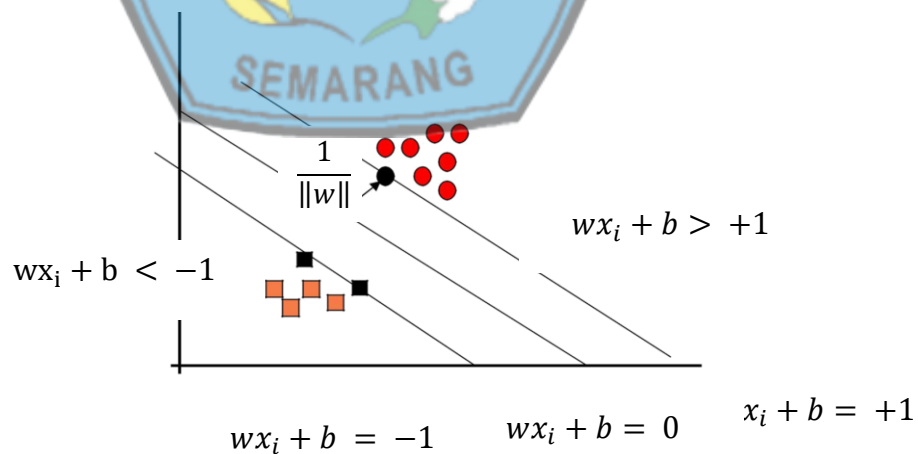
$$wx_1 + b = +1$$

$$wx_2 + b = -1$$

$$w(x_1 - x_2) = 2$$

$$\rightarrow \frac{w(x_1 - x_2)}{\|w\|} = \frac{2}{\|w\|}$$

Pada dasarnya jumlah fungsi pemisah ini tidak terbatas banyaknya. Misalkan dari jumlah yang tidak terbatas ambil dua, yaitu $f_1(x)$ dan $f_2(x)$ (lihat gambar 2.1). Fungsi f_1 mempunyai margin yang lebih besar dari pada fungsi f_2 . Setelah menemukan dua fungsi, suatu data baru masuk dengan keluaran -1 . Kemudian data tersebut akan dikelompokkan dalam kelas -1 atau $+1$ menggunakan fungsi pemisah yang sudah temukan.



Gambar 2. 3 Mencari fungsi pemisah yang optimal untuk obyek yang bisa dipisahkan secara linier.

Dengan menggunakan f_1 , akan dikelompokkan data baru di kelas -1 yang artinya adalah benar. Sedangkan f_2 akan di tempatkan di kelas $+1$ yang berarti salah.

Secara matematika, formulasi problem optimisasi SVM untuk kasus klasifikasi linier di dalam *primal space* adalah:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (8)$$

Subyek Untuk

$$y_i(wx_i + b) \geq 1, \quad i = 1, \dots, m$$

Dimana x_i adalah data input, y_i adalah *output* yang nilainya =1 atau -1, w dan b adalah parameter yang akan dicari nilainya. Dalam formulasi di atas, akan meminimalkan fungsi tujuan (*obyektif function*) $\frac{1}{2} \|w\|^2$, atau memaksimalkan kuantitas $\frac{1}{\|w\|}$ atau $w^T w$ dengan memperhatikan pembatas $y_i(wx_i + b) \geq 1$. Bila output data $y_i = +1$, maka pembatas menjadi $(wx_i + b) \geq 1$.

Sebaliknya bila $y_i = -1$, pembatas menjadi $(wx_i + b) \leq -1$. Di dalam kasus yang tidak *feasible* atau tidak mungkin dimana beberapa data mungkin tidak bisa dikelompokkan secara benar dan tepat, formulasi matematikanya menjadi berikut:

$$\min_{w,b,t} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m t_i \quad (9)$$

Subyek dari

$$y_i(wx_i + b) + t_i \geq 1 \quad t_i \geq 0 \quad i = 1, \dots, m$$

Dimana t_i adalah variabel *slack* (Haykin, 1999). Dalam persamaan diatas akan meminimalkan kesalahan klasifikasi (*misclassification error*) yang dinyatakan dengan adanya variabel *slack* t_i , sementara dalam waktu yang sama akan memaksimalkan margin untuk memisahkan kedua kelompok. Penggunaan variabel *slack* t_i adalah untuk mengatasi kasus ketidaklayakan (*infeasibility*) dari pembatas (*constraints*) $y_i(wx_i + b) \geq 1$ dengan cara memberi pinalti untuk data yang tidak memenuhi margin tersebut. Untuk meminimalkan nilai t_i ini, diberikan sebuah pinalti dengan menerapkan konstanta C . Vektor w tegak lurus terhadap fungsi pemisah: $wx + b = 0$. Konstanta b menentukan lokasi fungsi pemisah relatif terhadap titik asal (*origin*). Permasalahan yang akan timbul pada persamaan (9) adalah *program nonlinear*. Ini bisa dilihat dari fungsi tujuan (*objective function*) yang berbentuk kuadrat. Untuk menyelesaikannya, secara komputasi sulit dan perlu waktu lebih panjang. Untuk memecahkan permasalahan ini agar lebih mudah dan efisien untuk diselesaikan, masalah persamaan 9 akan transformasikan ke dalam dual *space*. Untuk itu, permasalahan pada persamaan (9) diubah menjadi fungsi *Lagrange*:

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^N \alpha_i [y_i(w^T x_i + b) - 1] \quad (10)$$

Dimana variabel *non-negatif* α_i , dinamakan *Lagrange multiplier*. Solusi dari problem optimisasi dengan pembatas seperti di atas ditentukan dengan mencari *saddle point* dari fungsi Lagrangian $J(w, b, \alpha)$. Fungsi ini harus diminimalkan terhadap variabel w dan b dan harus dimaksimalkan terhadap variabel α . Kemudian

dicari turunan pertama dari fungsi $J(w, b, \alpha)$ terhadap variabel w dan b sama dengan

0. Dengan melakukan proses ini akan mendapatkan dua kondisi optimalitas berikut:

a. Kondisi 1:

$$\frac{\partial J(w, b, \alpha)}{\partial w} = 0$$

b. Kondisi 2:

$$\frac{\partial J(w, b, \alpha)}{\partial b} = 0$$

Penerapan kondisi optimalitas 1 pada fungsi Lagrangian (10) akan menghasilkan

$$w = \sum_{i=1}^N \alpha_i y_i x_i \quad (11)$$

Penerapan kondisi optimalitas 2 pada fungsi Lagrangian (10) akan menghasilkan

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (12)$$

Menurut *duality theorem* (Bertsekas, 1995):

a. Jika problem primal mempunyai solusi optimal, maka problem dual juga akan mempunyai solusi optimal yang nilainya sama.

Bila w_0 adalah solusi optimal untuk problem primal dan α_0 untuk problem dual, maka perlu dan cukup bahwa w_0 solusi layak untuk problem primal:

$$\Phi(w_0) = J(w_0, b_0, \alpha_0) = \min_w J(w, b, \alpha) \quad (13)$$

Untuk mendapatkan problem dual, persamaan (4) dijabarkan sebagai berikut:

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^m \alpha_i y_i w^T x_i - b \sum_{i=1}^m \alpha_i y_i - \sum_{i=1}^m \alpha_i$$

Menurut kondisi optimalitas ke dua dalam persamaan (12), perlakuan ketiga sisi sebelah kanan dalam persamaan di atas sama dengan 0. Dengan memakai nilai-nilai w dalam persamaan (11), didapatkan

$$w^T w = \sum_{i=1}^m \alpha_i y_i w^T x_i = \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (14)$$

Maka persamaan 7 menjadi

$$Q(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (15)$$

Selanjutnya didapat formulasi dual dari problem (9):

$$\max \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (16)$$

Subjek dari

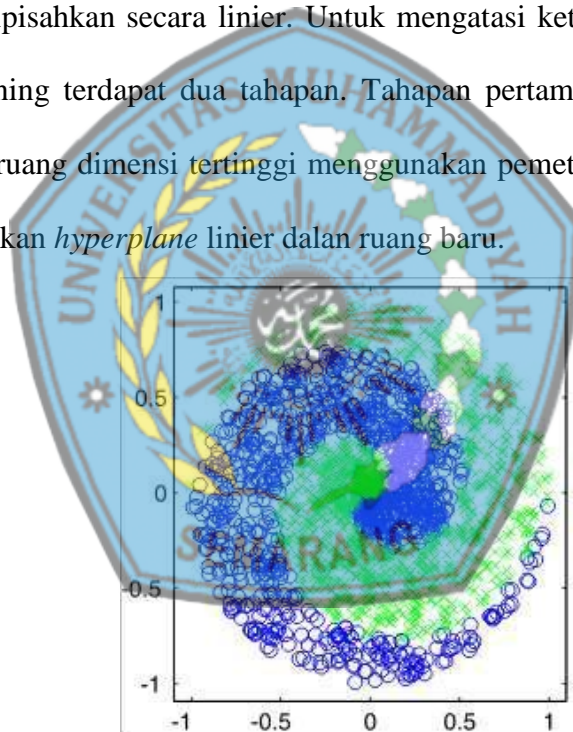
$$\sum_{i=1}^m \alpha_i y_i = 0$$

$$0 \leq \alpha_i, i = 1, \dots, m$$

Persamaan (14 dan 16) adalah *quadratic programming* (QP) dengan pembatas (*constraint*) linier. Karena itu solusi dari SVM adalah unik dan global optimal. Hal ini berbeda dengan solusi *neural networks* (Haykin, 1999) yang ekuivalen dengan problem *nonconvex optimization* dengan akibat solusi yang ditemukan adalah *local optima*.

2.5. Support Vector Machine Non-Linier

Banyak teknik data mining atau *machine learning* yang dikembangkan dengan asumsi kelinieran. Sehingga algoritma yang dihasilkan terbatas untuk kasus-kasus yang linier. Karena itu, bila suatu kasus klasifikasi memperlihatkan ketidak linieran, algoritma seperti *perceptron* tidak bisa mengatasinya. Secara umum, kasus-kasus di dunia nyata adalah kasus yang tidak linier. Sebagai contoh, perhatikan Gambar 2.2, data ini sulit dipisahkan secara linier. Untuk mengatasi ketidak linieran suatu data dalam data mining terdapat dua tahapan. Tahapan pertama, transformasikan data *input* kedalam ruang dimensi tertinggi menggunakan pemetaan non-linier. Tahapan kedua memisahkan *hyperplane* linier dalam ruang baru.



Gambar 2. 4 Data spiral yang menggambarkan ketidaklinieran

Namun pada umumnya untuk data yang distriusi kelasnya tidak linier biasanya digunakan pendekatan kernel pada awal set data. Kernel didefinisikan sebagai suatu fungsi yang memetakan fitur data dari dimensi awal (rendah) ke fitur lain yang berdimensi lebih tinggi (bahkan jauh lebih tinggi). Macam-macam jenis kernel yang sering digunakan dapat dilihat pada tabel 2.2. Pendekatan dengan kernel berbeda

dengan metode klasifikasi pada umumnya yang justru mengurangi dimensi awal untuk menyederhanakan proses komputasi dan memberikan akurasi prediksi yang lebih baik. Algoritma pemetaan kernel ditunjukkan pada Algoritma $\Phi: D^q \rightarrow D^r, x \rightarrow \Phi(x)$. Φ merupakan fungsi kernel yang digunakan untuk pemetaan, D merupakan data latih, dan q merupakan set fitur dalam satu data yang lama, dalam r merupakan set fitur yang baru sebagai hasil pemetaan untuk setiap data latih. Sementara x merupakan data latih, di mana $x_1, x_2, \dots, x_n \in D^q$ merupakan fitur-fitur yang akan dipetakan ke fitur berdimensi tinggi r , jadi untuk set data yang digunakan sebagai pelatihan dengan algoritma yang ada dari dimensi fitur yang lama D ke dimensi baru r . Misalnya, untuk n sampel data:

$$(\Phi(x_1), y_1, \Phi(x_2), y_2, \dots, \Phi(x_n), y_n) \in D^r \quad (17)$$

Pemetaan fitur lama pada set data ke fitur baru merupakan analogi dari layer tersembunyi pada ANN di mana jumlah neuron dalam layer tersembunyi biasanya lebih banyak daripada jumlah vektor masukan. Selanjutnya dilakukan proses pelatihan yang sama sebagaimana pada SVM linier. Proses pemetaan pada fase ini memerlukan perhitungan dot-product dua buah data pada ruang fitur baru. Dot-product buah vector (x_i) dan (x_j) dinotasikan sebagai $\Phi(x_i) \cdot \Phi(x_j)$. Nilai dot-product kedua buah vektor ini dapat dihitung secara tidak langsung, yaitu tanpa mengetahui fungsi transformasi Φ . Teknik komputasi seperti ini kemudian disebut trik kernel, yaitu menghitung *dot-product* dua buah vektor di ruang dimensi baru

dengan memakai komponen kedua buah vektor tersebut di ruang dimensi asal, seperti berikut:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (18)$$

Dan prediksi pada set data dengan dimensi fitur yang baru diformasikan dengan

$$f(\Phi(x)) = \text{sign}(w \cdot \Phi(z) + b) = \text{sign}\left(\sum_{i=1}^m \alpha_i y_i \Phi(x_i) \cdot \Phi(z) + b\right) \quad (19)$$

N adalah jumlah data yang menjadi *support vector*, x_i adalah *support vector*, dan z adalah data uji yang akan diprediksi:

Tabel 2. 2Macam-macam Kernel

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

x dan y adalah pasangan dua data dari semua bagian data latih. Paramater $\sigma, c, d > 0$, merupakan konstanta. Fungsi kernel mana yang harus digunakan untuk substitusi dot-product dari fitur dimensi lama ke dimensi baru sangat bergantung pada kondisi data (Santosa, 2007). Pemilihan fungsi kernel yang tepat merupakan hal yang sangat penting karena fungsi kernel akan menentukan fitur baru (dimensi tinggi) di

mana fungsi klasifikasi (*hyperplane*) akan dicari. Sepanjang fungsi kernel yang dipilih tepat maka SVM akan bekerja sesuai dan benar. Fungsi kernel yang sah diberikan oleh teori ([Vapnik, 1995] dan [Haykin, 1999]) dimana fungsi kernel harus memenuhi syarat kontinu dan pasti positif.

2.6. Smooth Support Vector Machine (SSVM)

Smooth Support Vector Machine (SSVM) merupakan pengembangan dari klasifikasi *Support Vector Machine* (SVM) dimana melibatkan fungsi *smoothing* didalamnya. Jika terpadat masalah dari klasifikasi titik m di dalam n ruang dimensi nyata R^n , gambaran dari matriks $m \times n$ adalah matrik A , anggota dari titik A_i di dalam kelas $+1$ atau -1 seperti yang diberikan $m \times m$ matriks diagonal D dengan diagonal $+1$ atau -1 . Untuk masalah *support vector machine* dengan sebuah *kernel* linier AA' diberikan untuk $v > 0$:

$$\min_{(w, \gamma, y) \in R^{n+1+m}} v e' y + \frac{1}{2} w' w \quad (19)$$

$$s. t \ d(Aw - e\gamma) + y \geq e$$

$$y \geq 0$$

Dimana w adalah normal untuk *bounding planes*:

$$x'w - \gamma = +1 \quad (20)$$

$$x'w - \gamma = -1$$

Dan γ menentukan daerah *relative* asal. Garis pemisah pertama batasnya adalah $+1$ dan garis pemisah kedua batasnya adalah -1 ketika kedua kelas

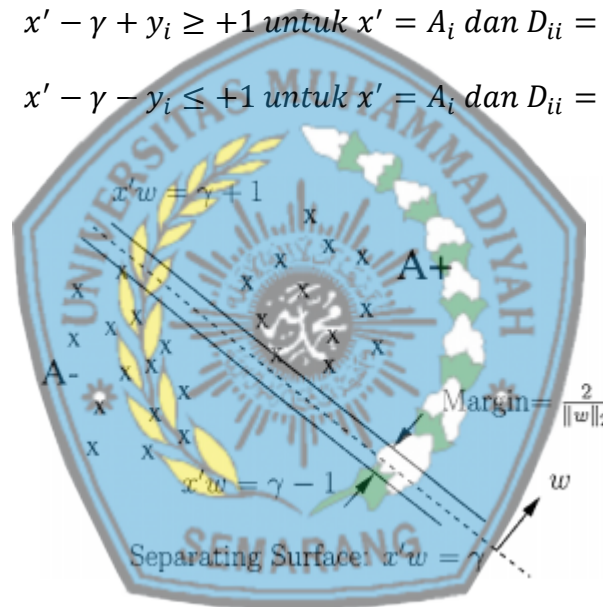
dipisahkan secara *linier*, yaitu ketika variabel *slack* $y = 0$. Garis pemisah linier ketika:

$$x'w = \gamma \quad (21)$$

Batasan tengah antara dua garis (lihat gambar 2.4). Jika kelas linier tidak bisa dipisahkan kemudian dua palne membatasi dua kelas dengan "soft margin" ditentukan oleh variabel *slack* y nonnegatif, yaitu

$$x' - \gamma + y_i \geq +1 \text{ untuk } x' = A_i \text{ dan } D_{ii} = +1 \quad (22)$$

$$x' - \gamma - y_i \leq +1 \text{ untuk } x' = A_i \text{ dan } D_{ii} = -1$$



Gambar 2. 5kedua garis pembatas dengan margin $\frac{2}{\|w\|_2}$ dan garis ke-3 memisahkan $A +$, titik tersebut mewakili dari baris A dengan $D_{ii} = +1$. Dari $A -$, titik tersebut mewakili A dengan $D_{ii} = -1$.

Aturan pertama dari variabel *slack* y adalah meminimaliasi dengan pembobot v dalam persamaan (19). Hubungan kuadrat dalam persamaan (19), yang kedua adalah timbal balik jarak $\frac{2}{\|w\|_2}$ diantara dua garis pembatas dalam n -ruang dimensi dari $w \in R^n$ untuk sebuah , maksimal jarak, seringkali disebut "margin". Gambar 2.3

menggambarkan perwakilan titik A , pembatas kedua garis dengan margin $\frac{2}{\|\omega\|_2}$, dan pemisah garis 3 yang memisahkan $A +$, mewakili titik dari baris A dengan $D_{ii} = 1$, dari $A -$, mewakili titik dari baris A dengan $D_{ii} = -1$.

$$\min_{\omega, \gamma, y} \frac{\nu}{2} y' y + \frac{1}{2} (w + \gamma^2) \quad (23)$$

$$s. t \ D(Aw - e\gamma) + y \geq e$$

$$y \geq 0$$

Solusi dari permasalahan di atas adalah

$$y = (e - D(Aw - e\gamma))_+ \quad (24)$$

Dimana, terlebih dahulu menggambarkan, $(.)_+$ diganti dengan komponen *negative* dari vektor nol, dengan demikian dapat mengganti y dalam persamaan (23) dengan $(e - D(Aw - e\gamma))_+$ dan mengubah masalah *Support Vector Machine* (SVM) pada persamaan (23) kedalam *Support Vector Machine* (SVM) yang merupakan masalah optimasi dibatasi sebagai berikut:

$$\min_{\omega, \gamma} \frac{\nu}{2} \|(e - D(A\omega - e\gamma))_+\|_2^2 + \frac{1}{2} (\omega' \omega + \gamma^2) \quad (25)$$

Masalah ini adalah masalah minimisasi tanpa banyak kendala. Tetapi tujuan fungsi dalam persamaan (25) tidak dua kali terdiferensiasi yang menghalangi penggunaan metode *Newton*. akan diterapkan teknik *smoothing* dan mengganti x_+ dengan sebuah akurasi penaksiran *smooth* pemulusan (lihat **Lemma 2.1**) yang diberikan oleh $p(x, \alpha)$, fungsi integral sigmoid $\frac{1}{1 + e^{-\alpha x}}$ dari neural networks,

$$p(x, \alpha) = x + \frac{1}{\alpha} \log(1 - \varepsilon^{-\alpha x}), \alpha > 0 \quad (26)$$

Fungsi p dengan parameter pemulusan α digunakan untuk menggantikan fungsi $plus$ pada persamaan (25) untuk mendapatkan sebuah *Smooth Support Vector Machine* (SSVM)

$$\min_{(\omega, \gamma) \in \mathbb{R}^{n+1}} \Phi_{\alpha}(\omega, \gamma) := \min_{(\omega, \gamma) \in \mathbb{R}^{n+1}} \frac{\nu}{2} \|p(e - D(A\omega - e\gamma), \alpha)\|_2^2 + \frac{1}{2}(\omega' + \gamma^2). \quad (27)$$

Akan diberikan sebuah solusi untuk pemecahan permasalahan (23) yaitu persamaan (27) dengan α mendekati tak hingga. Terdapat suatu keuntungan dari sifat dua kali terdeferensiasi dari fungsi tujuan dalam persamaan (27) untuk memanfaatkan algoritma konvergen kuadratik *Smooth Support Vector Machine* (SSVM).

Dimulai dengan *lemma* sederhana yang batas-batas perbedaan antara $plus(x)$ + dan pendekatan pemulusan $p(x, \alpha)$.

Lemma 2.1 untuk $x \in \mathbb{R}$ dan $|x| \leq \rho$: $p(x, \alpha)^2 - (x_+)^2 \leq \left(\frac{\log 2}{\alpha}\right)^2 + \frac{2\rho}{\alpha} \log 2$ dimana $p(x, \alpha)$. Adalah fungsi p dari (8) dengan pemulusan parameter $\alpha > 0$

Bukti: Mempertimbangkan dua kasus. Untuk $0 < x < \rho$

$$p(x, \alpha)^2 - (x_+)^2 = \frac{1}{\alpha^2} \log(1 + \varepsilon^{-\alpha x}) + \frac{2x}{\alpha} \log(1 + \varepsilon^{-\alpha x}) \leq \left(\frac{\log 2}{\alpha}\right)^2 + \frac{2\rho}{\alpha} \log 2$$

Untuk $-\rho < x \leq 0$, $p(x, \alpha)^2$ adalah fungsi naik yang monoton, jadi:

$$p(x, \alpha)^2 - (x_+)^2 = p(x, \alpha)^2 \leq p(0, \alpha)^2 = \left(\frac{\log 2}{\alpha}\right)^2$$

Karena $p(x, \alpha)^2 - (x_+)^2 < \left(\frac{\log 2}{\alpha}\right)^2 + \frac{2\rho}{\alpha} \log 2$.

Sekarang akan menunjukkan bahwa parameter *smoothing* α mendekati tak hingga solusi yang khusus untuk masalah pemulusan pada persamaan (27) pendekatan khusus untuk solusi masalah yang setara dengan *Support Vector Machine* (SVM) (25) fungsi $f(x)$ diberikan dalam persamaan (28) yang menggolongkan fungsi setara *Support Vector Machine* (SVM) dari persamaan (25) dan fungsi $g(x, \alpha)$ diberikan dalam persamaan (29) yang tergolong fungsi *Smooth Support Vector Machine* (SSVM) pada persamaan (27).

Teorema 2.2 memberikan $A \in R^{m \times n}$ dan $b \in R^m$ tentukan fungsi nyata dari $f(x)$ dan $g(x, \alpha)$ di dalam n-dimensi ruang dimensi nyata R^n :

$$f(x) = \frac{1}{2} \|(Ax - b)_+\|_2^2 + \frac{1}{2} \|x\|_2^2 \quad (28)$$

Dan

$$g(x, \alpha) = \frac{1}{2} \|p((Ax - b), \alpha)\|_2^2 + \frac{1}{2} \|x\|_2^2 \quad (29)$$

Dengan $\alpha > 0$

(i) Ada sebuah solusi yang khusus \bar{x} dari $\min_{x \in R^n} f(x)$ dan solusi khusus untuk

\bar{x} dari $\min_{x \in R^n} g(x, \alpha)$.

(ii) Untuk semua $\alpha > 0$

$$\|\bar{x}_\alpha - \bar{x}\|_2^2 \leq \frac{m}{2} \left(\left(\frac{\log 2}{\alpha} \right)^2 + 2\xi \frac{\log 2}{\alpha} \right) \quad (30)$$

Dimana ξ dapat dicari dengan

$$\xi = \min_{1 \leq i \leq m} |(A\bar{x} - b)_i|. \quad (31)$$

\bar{x}_α konvergen untuk \bar{x} sama dengan α mendekati tak hingga dengan batas atas diberikan pada persamaan (25).

Pembuktian

- (i) Menunjukkan adanya solusi khusus, kita ketahui bahwa $x_+ \leq p(x, \alpha)$ tingkat ketetapan $L_v(g(x, \alpha))$ dan $L_v(f(x))$ adalah tetap.

$$L_v(g(x, \alpha)) \subseteq \{x \mid \|x\|_2^2 \leq 2v\}, \quad (32)$$

- (ii) Untuk $\alpha > 0$. sebab $L_v(g(x, \alpha))$ dan $L_v(f(x))$ sub set yang padat dalam R^n dan masalah $\min_{x \in R^n} f(x)$ dan $\min_{x \in R^n} g(x, \alpha)$ mempunyai solusi/pemecahan.

- (iii) Untuk membangun konvergensi, kami mencatat bahwa dengan kondisi optimasi pertama dari $f(x)$ dan $g(x, \alpha)$.

$$f(\bar{x}_\alpha) - f(\bar{x}) \geq \nabla f(\bar{x})(\bar{x}_\alpha - \bar{x}) + \frac{1}{2} \|\bar{x}_\alpha - \bar{x}\|_2^2 = \frac{1}{2} \|\bar{x}_\alpha - \bar{x}\|_2^2, \quad (33)$$

$$g(\bar{x}, \alpha) - g(\bar{x}_\alpha, \alpha) \geq \nabla g(\bar{x}_\alpha, \alpha)(\bar{x}_\alpha - \bar{x}) + \frac{1}{2} \|\bar{x}_\alpha - \bar{x}\|_2^2 = \frac{1}{2} \|\bar{x}_\alpha - \bar{x}\|_2^2 \quad (34)$$

Karena p mendominasi fungsi dan ditambah mempunyai fungsi $g(x, \alpha) - f(x) \geq 0$ untuk semua $\alpha > 0$ menambahkan (33) dan (34) dan menggunakan

$$\begin{aligned} \|\bar{x}_\alpha - \bar{x}\|_2^2 &\leq (g(\bar{x}, \alpha) - f(\bar{x})) - (g(\bar{x}_\alpha, \alpha) - f(\bar{x}_\alpha)) \leq g(\bar{x}, \alpha) - f(\bar{x}) \\ &= \frac{1}{2} \|p((A\bar{x} - b), \alpha)\|_2^2 - \frac{1}{2} \|(A\bar{x} - b)_+\|_2^2 \end{aligned} \quad (35)$$

Penerapan dari **Lemma 2.1** memberikan

$$\|\bar{x}_\alpha - \bar{x}\|_2^2 \leq \frac{m}{2} \left(\left(\frac{\log 2}{\alpha} \right) + 2\xi \frac{\log 2}{\alpha} \right), \quad (36)$$

Dimana ξ adalah angka positif didefinisikan dalam persamaan (31). Sarat dalam persamaan (36) akan konvergen ke nol sebagai α sampai tak hingga. \bar{x}_α konvergen terhadap \bar{x} sebagai α menuju tak hingga dengan batas atas diberikan dalam persamaan (27).

2.7. Algoritma Newton Armijo untuk *Smooth Support Vector Machine*

Dengan memanfaatkan hasil bagian sebelumnya dan mengambil keuntungan dari dua kali perbedaan dari fungsi objective dari masalah persamaan (27), akan dilakukan penyusunan algoritma *Newton* yang konvergen dengan langkah Armijo.

Algoritma 2.1 (Algoritma Newton-Armijo untuk SSVM (27), mulai dengan $(w^0, \gamma^0) \in R^{n+1}$ setelah (w^i, γ^i) , berhenti jika kemiringan dari fungsi objektif (27) adalah nol, itu adalah $\nabla\Phi_\alpha(w^i, \gamma^i) = 0$, dengan langkah sebagai berikut:

- (i) Petunjuk *Newton*: Petunjuk determinan $d^i \in R^{n+1}$ dengan keadaan linier dari $\nabla\Phi_\alpha(w, \gamma)$ di sekitar (w^i, γ^i) yang memberikan persamaan $n + 1$ linier di dalam $n + 1$ variabel:

$$\nabla^2\Phi_\alpha(w^i, \gamma^i)d^i = -\nabla\Phi_\alpha(w^i, \gamma^i)' \quad (37)$$

- (ii) Langkah Armijo [1]: pilih sebuah langkah $\lambda_i \in R$ sedemikian

$$(w^{i+1}, \gamma^{i+1}) = (w^i, \gamma^i) + \lambda_i d^i \quad (38)$$

Dimana $\lambda_i = \max\left\{1, \frac{1}{2}, \frac{1}{4}, \dots\right\}$ sedemikian hingga

$$\Phi_\alpha(w^i, \gamma^i) - \Phi_\alpha(w^i, \gamma^i) + \Phi_\alpha((w^i, \gamma^i) + \lambda_i d^i) \geq -\delta\lambda_i\Phi_\alpha(w^i, \gamma^i)d^i \quad (39)$$

Dimana $\delta \in (0, \frac{1}{2})$

Perbedaan utama antara pendekatan *smoothing* dan *Support Vector Machine* (SVM) adalah pemecahan sistem linier (37). Sebagai alternative pemecahan masalah kuadratik dengan keadaan SVM klasik. Selanjutnya kita dapat menampilkan Algoritma *smoothing* konvergen global untuk solusi yang unik dan mengambil langkah *Newton* setelah sejumlah *afinite* iterasi. Hal ini menyebabkan hasil konvergensi kuadrat berikut:

Teorema 2.1 memasukan $\{(w^i, \gamma^i)\}$ akan menghasilkan sebuah rangkaian Algoritma 2.1 dan $(\bar{w}, \bar{\gamma})$ menjadi sebuah solusi khusus untuk masalah persamaan(27).

- (i) Rangkaian $\{(w^i, \gamma^i)\}$ konvergen untuk solusi khusus $(\bar{w}, \bar{\gamma})$ titik awal (w^0, γ^0) di dalam R^{n+1}

Untuk titik awal (w^0, γ^0) terdapat bilangan \bar{i} sehingga langkah λ_i dari Algoritma 2.1, 1 untuk $i \geq \bar{i}$ dan rangkaian $\{(w^i, \gamma^i)\}$ konvergen untuk $(\bar{w}, \bar{\gamma})$ kuadratik.

2.8. Smooth Support Vector Machine dengan Non-linier Kernel

Formulasi *Smooth Support Vector Machine* (SSVM) pada dasarnya adalah linier untuk masalah klasifikasi. Permasalahan muncul ketika kasus tidak linier. cara melinierkan kasus tidak linier dengan menggeneralisasi menggunakan kernel. Dengan pemecahannya generalisasi dengan *kernel* $K(A, A')$ sebagai berikut:

$$\min_{\omega, \gamma, y} v e' y + f(u) \quad (40)$$

$$s. t D(K(A, A')Du - e\gamma) + y \geq e$$

$$y \geq e$$

$f(u)$ adalah fungsi cembung dalam R^m yang mana parameter u dan v angka positif yang bobot kesalahan klasifikasi $e'y$. Sebuah solusi matematik untuk u dan γ pengaruh non-linier:

$$K(x', A')Du = \gamma \quad (41)$$

Formulasi *linier* (32) dari bagian 2 didapat jika $K(A, A') = AA'$, $\omega = A'Du$ dan $f(u) = \frac{1}{2}u'DAA'Du$. Sebuah perbedaan objektifitas kalsifikasi tidak hanya menekankan parameter u tetapi juga menekankan γ didalam formulasi non-linier:

$$\min_{\omega, \gamma, y} \frac{v}{2} y'y + \frac{1}{2} (u'u + \gamma^2) \quad (42)$$

$$\text{s. t } D(K(A, A')DU - e\gamma) + Y \geq e$$

$$y \geq 0.$$

SSVM dengan non-linier kernel $K(A, A')$:

$$\min_{u, v} \frac{v}{2} \|p(e - D(k(A, A')Du - e\gamma), \alpha)\|_2^2 + \frac{1}{2} (u'u + \gamma^2) \quad (43)$$

Dimana $K(A, A')$ adalah sebuah kernel dari $R^{m \times n} \times R^{n \times m} R^{m \times m}$. Catatan penting untuk masalah ini, yang mampu menghasilkan pemisah yang sangat nonlinear, masih mempertahankan kecembungan dan ketidak samaan sifat untuk kernel yang berubah-ubah. Semua hasil dari bagian sebelumnya. Kita dapat menerapkan Algoritma Newton-Armijo untuk pemecahan langsung.

2.9. K-fold cross validation (KCV)

K-fold cross validation merupakan teknik untuk membagi dokumen menjadi k bagian. Pembagian tersebut merupakan pembagian sebagai data *training* dan *test set*.

Seluruh data secara acak dibagi menjadi K buah subset B_k dengan ukuran yang sama dimana himpunan bagian dari $\{1, \dots, n\}$, sedemikian sehingga $\bigcup_{k=1}^K B_k = \{1, \dots, n\}$ dan $B_j \cap B_k = \emptyset (j \neq k)$. Kemudian dilakukan tahap iterasi sebanyak k kali, kemudian pada iterasi ke k subset B_k menjadi *test test* sedangkan sisanya menjadi *training set*. Kelebihan dari metode *k-fold cros validation* adalah tidak adanya permasalahan dalam pembagian data.

2.10. Pengukuran Kinerja Klasifikasi

Diharapkan dalam suatu klasifikasi semua data dapat diklasifikasi dengan benar, tetapi terkadang tidak bisa diklasifikasi yang didapat tidak mencapai 100% benar sehingga sebuah sistem klasifikasi juga harus diukur kinerjanya. Dalam pengukuran kinerja klasifikasi umumnya dilakukan dengan matriks konfusi (*confusion matrix*).

Pada tabel 2.3 Merupakan contoh dari tabel konfusi untuk mengukur hasil kerja klasifikasi dengan masalah 2 kelas (*biner*). Hanya ada dua kelas yaitu kelas 0 dan kelas 1. Setiap f_{ij} dalam matriks menyatakan jumlah *record*/ data dari kelas i yang prediksinya masuk kedalam kelas j .

Tabel 2. 3 Matriks Konfusi untuk Dua Kelas(Prasetyo, 2012)

f_{ij}	Kelas hasil prediksi(j)		
	Kelas = 1	Kelas = 0	
Kelas asli(i)	Kelas = 1	Kelas = 0	
	f_{11}	f_{10}	
	Kelas = 0	f_{01}	f_{00}

Dalam tabel 2.3 merupakan kasus dengan dua kelas atau klasifikasi biner namun jika terdapat 4 kelas dapat ditulis sebagai berikut:

Tabel 2. 4 Matriks Konfusi Empat Kelas

f_{ij}		Kelas hasil prediksi(j)			
		Kelas = 1	Kelas = 2	Kelas =3	Kelas = 4
Kelas asli(i)	Kelas = 1	f_{11}	f_{12}	f_{13}	f_{14}
	Kelas = 2	f_{21}	f_{22}	f_{23}	f_{24}
	Kelas = 3	f_{31}	f_{32}	f_{33}	f_{34}
	Kelas = 4	f_{41}	f_{42}	f_{34}	f_{44}

Berdasarkan isian dari matriks konfusi dapat dihitung jumlah data yang benar dan jumlah data yang salah. Kuantitas matriks konfusi dapat dihitung akurasi dan laju *error*. Setelah mengetahui jumlah kasifikasi yang benar, dapat diketahui akurasi hasil prediksi. Sedangkan untuk hasil klasifikasi yang salah dapat kita hitung laju error dari prediksi yang dilakukan. Akurasi dan laju *error* digunakan sebagai tolak ukur kinerja kasifikasi.

Untuk mengitung hasil laju *error* dan akurasi menggunakan formula (Prasetyo, 2012)

$$\text{Akurasi} = \frac{\text{jumlah data yang diprediksi secara benar}}{\text{jumlah prediksi yang dilakukan}} \times 100\% \quad (44)$$

$$\text{Laju error} = \frac{\text{jumlah data yang diprediksi secara salah}}{\text{jumlah prediksi yang dilakukan}} \times 100\% \quad (45)$$

Model yang dibangun dapat memprediksi dengan benar pada semua data yang menjadi semua data latihnya, namun saat model perhadapan dengan data uji, disaat itu juga kinerja model dari sebuah algoritma klasifikasi ditentukan (Prasetyo, 2012).