BAB II
LANDASAN TEORI

1.1. Tinjauan Pustaka

Dari hasil pencarian pada berbagai sumber penelitian karya ilmiah, terdapat beberapa penelitian yang berkaitan dengan analisis sentimen antara lain:

Dedi Darwis, et al., (2020) dalam pembahasannya yang berjudul Penerapan Algoritma SVM untuk Analisis Sentimen pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia menggunakan strategi Support VectorMachine. Akibat dari urutan pemanfaatan teknik SVM tersebut dipisahkan menjadi tiga kelas, yaitu kelas positif 8%, kelas negatif 77%, dan kelas tidak bias 15%. Uji coba pemeriksaan ini menggunakan Confusion Matrix, ketepatan hasil eksplorasi secara umum akan menghasilkan opini negatif sebesar 77% dengan ketepatan hasil eksperimen ketepatan 82% dan akurasi pengujian 90%, serta review dari 88% dan skor f1 sebesar 89%.

Debby Alita, et al., (2020) dalam eksplorasi Implementasi Algoritma Multiclass SVM dalam Opini Publik Bahasa Indonesia di Twitter dengan SVM One Against One dan SVM One Against Rest dengan konsekuensi SVM One Against One tidak tertandingi untuk nilai akurasi, tinjauan, dan F1Score, sedangkan untuk ketepatan SVM One Against Rest menonjol dengan selisih senilai 0,06. Ketika dikontraskan dan dibeda-bedakan strateginya, khususnya strategi pengelompokan Naïve Bayes Classifier dalam proses investigasi opini, teknik multiclass SVM sebenarnya memiliki nilai yang tak tertandingi dan

prosedur dataset yang dilakukan secara serampangan memiliki nilai yang lebih unggul.

Styawati, et al., (2021) dalam ujiannya yang berjudul Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja di Twitter Dengan Metode Support Vector Machine. Dalam ulasan ini, pemeriksaan dua bit selesai, khususnya lurus dengan RBF. Hasil penilaian yang diselesaikan pada ketepatan bit langsung senilai 98,67%, akurasi 98%, review mendekati 100%, dan F1-Score 98%, sedangkan nilai ketepatan bit RBF adalah 98,34%, akurasi 97%, review 98% , F1- Score 98%, dapat diduga bahwa perasaan publik dari klien twitter terhadap program pra-kartu nama selama pandemi lebih cenderung non-partisan sebesar 98,34%. Mengingat efek lanjutan dari penilaian yang dilakukan pada nilai presisi bit langsung, itu menciptakan nilai ketepatan 98,67%, sedangkan porsi RBF memberikan presisi 98,34%. Sejauh presisi, potongan lurus lebih tepat daripada bagian RBF.

Penelitian Ade Iriani, dkk. (2020), melakukan penelitian analisis sentiment terhadap transfortasi *online* pada media sosial *twitter* menggunakan algoritma *Support Vector Machine* (SVM) Berbasis *Particle Swarm Optimization* (PSO), hasil penelitian ini menghasilkan Analisis sentimen positif menggunakan SVM adalah sebesar 62% dan sentimen negatif sebesar 38%, sedangkan pada SVM- PSO, opini positif sebesar 53% dan negatif 47%. Hasil penelitian menggunakan 10 k-fold CV menghasilkan akurasi pada SVM sebesar 95,46% dan AUC 0,979

(*excellent classification*), sedangkan pada SVM-PSO sebesar 96,04% dan AUC 0,993 (*excellent classification*).

Penelitian Lilyani, dkk (2017), melakukan penelitian analisis sentiment terhadap opini publik terhadap kebakaran hutan pada media social melalui komparasi algoritma *Support Vector Machine* dan k-*Nearest Neighbor* berbasis *Particle Swarm Optimization.* Algoritma perbandingan SVM menghasilkan akurasi 80,83% dan AUC 0,947, kemudian dibandingkan dengan SVM berbasis PSO dengan akurasi 87,11% dan AUC 0,922. Data hasil pengujian untuk algoritma K-NN akurasinya adalah 85,00% dan AUC 0,918, kemudian dibandingkan untuk akurasi K-NN berbasis PSO sebesar 73,06% dan AUC 0.500.

Penelitian Kiki Setiawan, dkk. (2020), melakukan penelitian analisis sentiment terhadap mobile esemka mengunakan komparasi metode *Naive Bayes* dan *Support Vector Machine* menggunakan *Particle Swarm Optimization,* mendapatkan hasil akurasi sebesar 75.04% untuk nilai akurasi *Naive Bayes* tanpa seleksi fitur, 83.33% untuk hasil *Naive Bayes* menggunakan seleksi fitur. kemudian nilai akurasi yang didapat untuk *Support Vector Machine* tanpa seleksi fitur sebesar 78.81%, dan dengan fitur seleksi sebesar 88.19%.

Penelitian Elly Indrayuni, dkk (2016), melakukan penelitian Analisa Sentimen Review Hotel Menggunakan Algoritma *Support Vector Machine* Berbasis *Particle Swarm Optimization*, yang Hasil penelitian menunjukkan peningkatan nilai akurasi sebesar 5.61% untuk algoritma *Support Vector*

Machine dari 91.33% menjadi 96.94% setelah penerapan seleksi fitur *Particle Swarm Optimization*.

Penelitian Anas Faisal, dkk (2020), melakukan penelitian terhadap Analisis Sentimen Dewan Perwakilan Rakyat Dengan Algoritma Klasifikasi Berbasis *Particle Swarm Optimization*, Penelitian dilakukan dengan menggunakan dua Algoritma yaitu Algoritma SVM dan *Naive Bayes* dioptimasi menggunakan *Particle Swarm Optimization* (PSO). Hasil pengujian k-*fold cross validation* SVM dan NB mendapatkan nilai accuracy 71,04% dan 70,69% dengan nilai Area *Under the Curve* (AUC) 0,817 dan 0,661. Sedangkan hasil pengujian k-*flod cross validation* dengan menggunakan PSO, untuk SVM dan NB masing-masing mendapatkan nilai *accuracy* 75,03% dan 73,49% dengan nilai AUC 0,808 dan 0,719. Penggunaan PSO mampu meningkatkan nilai accuracy algoritma SVM sebesar 3,99% dan 2,8% pada algoritma NB. Hasil dari pengujian kedua algoritma tersebut nilai *accuracy* tertinggi adalah SVM dengan PSO sebesar 75,03%.

Tabel 2.1 Kajian pustaka.

No	Peneliti dan tahun peneliti	Judul artikel	Metode	Hasil/keterangan
1	Ade Iriani,dkk (2020),	Analisis Sentimen Transportasi Online Menggunakan *Support Vector Machine* Berbasis PSO	*Support Vector Machine* (SVM) Berbasis *Particle Swarm Optimization* (PSO)	sentimen positif menggunakan SVM sebesar 62% , sentimen negatif 38%, sedangkan pada SVM-PSO, opini positif 53% dan negatif 47%.

Tabel 2.1 Kajian pustaka lanjutan.

2.	Lilyani, dkk (2017)	Analisis sentimen opini publik berita kebakaran hutan Melalui komparasi algoritma *support vector Machine* dan k-*nearest neighbo*r berbasis *Particle swarm optimization*	Komparasi algoritma *support vector Machine* dan k-*nearest neighbor* berbasis *Particle swarm optimization*	akurasi 80,83% dan AUC 0,947, SVM + PSO = 87,11% dan AUC 0,922. Data hasil algoritma K-NN akurasinya adalah 85,00% dan AUC 0,918,K-NN + PSO = 73,06% dan AUC 0.500
3	Kiki Setiawan, dkk. (2020)	Komparasi Metode *Naive Bayes* Dan *Support Vector Machine* Menggunakan *Particle Swarm Optimization* Untuk Analisis Sentimen Mobil Esemka	Komparasi Metode *Naive Bayes* Dan *Support Vector Machine* berbasis *Particle Swarm Optimization*	75.04% akurasi Naive Bayes, 83.33% hasil Naive Bayes +PSO. Support Vector Machine 78.81%, dan dengan fitur seleksi PSO sebesar 88.19%
4	Dedi Darwis, et al (2020)	Penerapan Algoritma SVM untuk Analisis Sentimen pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia menggunakan strategi Support Vector Machine	Algoritma Support Vector Machine	opini negatif sebesar 77% dengan ketepatan hasil eksperimen ketepatan 82% dan akurasi pengujian 90%, serta review dari 88% dan skor f1 sebesar 89%.

Tabel 2.1 Kajian pustaka lanjutan.

5	Debby Alita, et al., (2020)	Implementasi Algoritma Multiclass SVM dalam Opini Publik Bahasa Indonesia di Twitter.	SVM One Against One dan SVM One Against Rest	SVM One Against One tidak tertandingi untuk nilai akurasi, tinjauan, dan F1Score, sedangkan untuk ketepatan SVM One Against Rest menonjol dengan selisih senilai 0,06.
6	Styawati, et al., (2021)	Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja di Twitter Dengan Metode Support Vector Machine	Metode Support Vector Machine	ketepatan bit 98,67%, akurasi 98%, review mendekati 100%, dan F1-Score 98%, ketepatan bit RBF adalah 98,34%, akurasi 97%, review 98% , F1-Score 98%,
7	Andi Nurkholis, et al., (2021)	Optimasi Parameter Support Vector Machine Berbasis Algoritma Firefly	algoritma Support Vector Machine Berbasis Algoritma Firefly	presisi tertinggi 87,84%. Penilaian berikut menggunakan cakupan nilai C=1.0-3.0 dan =1.0-2.0 menghasilkan presisi tertinggi 87,15%.
8	Sucitra Sahara,dkk (2020)	analisa review software antivirus dengan menggunakan metode K-Nearest Neighbors berbasis Particle Swarm Optm.	metode K-Nearest Neighbors berbasis Particle Swarm Optimization,	Algoritma k-NN yang diperoleh 70,50% , metode k-NN dan optimasi PSO didapatkan nilai akurasi sebesar 83,50%.

Tabel 2.1 Kajian pustaka lanjutan.

9	Elly Indrayuni,dkk (2016)	Analisa Sentimen Review Hotel Menggunakan Algoritma *Support Vector Machine* Berbasis *Particle Swarm Optimization*	Algoritma *Support Vector Machine* Berbasis *Particle Swarm Optimization*	Hasil nilai akurasi sebesar 5.61%, algoritma *Support Vector Machine* dari 91.33% menjadi 96.94% setelah penerapan seleksi fitur *Particle Swarm Optimization*
10	Anas Faisal,dkk (2020)	Analisis Sentimen Dewan Perwakilan Rakyat Dengan Algoritma SVM dan NB Berbasis *Particle Swarm Optimization*	Algoritma SVM dan NB Berbasis *Particle Swarm Optimization*	Hasil pengujian SVM dan NB = 71,04% dan 70,69% dengan nilai Area Under the Curve (AUC) 0,817 dan 0,661. Hasil pengujian menggunakan PSO, untuk SVM dan NB = 75,03% dan 73,49% dengan nilai AUC 0,808 dan 0,719.

1.2. Natural Lengguage Processing

Natural Language Processing (NLP) adalah sebuah proses untuk mengekstraksi teks bebas untuk mengetahui maknanya secara menyeluruh. Dalam NLP menggunakan teknik lingusitik yaitu *Part-of-Speech* dan struktur gramatikal. *Part-of-Speech* terdiri dari kata benda (*noun*), kata sifat (*adjective*), kata kerja (*verb*) dan lain lain, sedangkan struktur gramatikal tersusun atas ungkapan seperti ungkapan preposisi atau ungkapan kata benda atau hubungan salingketergantungan seperti subjek dari atau objek Kao & Poteet, (2007). Pustejovsky

& Moszkowicz, (2012) menjelaskan bahwa ada beberapa area utama penelitian pada area NLP, diantaranya:

1. *Question Answering Systems* (QAS) Kemampuan komputer untuk menjawab pertanyaan yang diberikan oleh user. Daripada memasukkan keyword ke dalam browser pencarian, dengan QAS, *user* bisa langsung bertanya dalam bahasa natural yang digunakannya, baik itu Inggris, Mandarin, ataupun Indonesia.
2. *Summarization* Pembuatan ringkasan dari sekumpulan konten dokumen atau *email*. Dengan menggunakan aplikasi ini, user bisa dibantu untuk mengkonversikan dokumen teks yang besar ke dalam bentuk *slide presentasi*.
3. *Machine Translation* Produk yang dihasilkan adalah aplikasi yang dapat memahami bahasa manusia dan menterjemahkannya ke dalam bahasa lain. Termasuk di dalamnya adalah *Google Translate* yang apabila dicermati semakin membaik dalam penterjemahan bahasa. Contoh lain lagi adalah *BabelFish* yang menterjemahkan bahasa pada real time.
4. *Speech Recognition Field* ini merupakan cabang ilmu NLP yang cukup sulit. Proses pembangunan model untuk digunakan telpon/komputer dalam mengenali bahasa yang diucapkan sudah banyak dikerjakan. Bahasa yang sering digunakan adalah berupa pertanyaan dan perintah.
5. *Document classification* Aplikasi ini adalah merupakan area penelitian NLP Yang paling sukses. Pekerjaan yang dilakukan aplikasi ini adalah menentukan dimana tempat terbaik dokumen yang baru diinputkan ke

dalam sistem. Hal ini sangat berguna pada aplikasi *spam filtering, news article classification, dan movie review.*

1.3. Text Mining

Text mining adalah proses penemuan pola ayang sebelumnya tidak terlihat pada dokumen atau sumber tertentu menjadi pola yang diinginkan untuk tujuan tertentu Feldman et al., (2009) *Text mining* sering digunakan untuk *analisis informasi,* pengambilan keputusan, dan tugas-tugas *manajemen informasi* lainnya yang terkait dengan data teks dalam jumlah besar.

Text mining merupakan perkembangan dari disiplin ilmu komputer dalam menangani masalah tentang *Natural Language*. *Text mining* mengadopsi beberapa metodologi dan teknik dari beberapa area yaitu *Information Retrieval, Information Extraction,* dan komputasi *linguistik* berbasis *corpus* Feldman et al.,(2009).

Sistem memanfaatkan peningkatan jumlah data yang tidak terstruktur dalam bentuk teks. Data tersebut diolah untuk memenuhi kebutuhan informasi menggunakan berbagai metode *klasifikasi*, *clustering,* analisis sentimen, dll.

Text mining melakukan beberapa tahapan proses dalam mengambil sebuah informasi. Tahapan proses yang dilakukan dalam *text mining* adalah sebagai berikut Kumar & Bhatia, (2013):

1. Text Preprocessing

Text Preprocessing merupakan pemrosessan yang dilakukan untuk membuat dokumen atau *text* agar lebih konsisten saat melakukan proses selanjutnya. Proses yang biasanya dilakukan adalah *text cleanup, tokenization* dan *part of*

speech tagging.

2. Text Transformation (Attribute Generation)

Text Transformation merupakan Proses yang dilakukan dengan mengambil *fitur* dari sebuah *text* dengan menggunakan pendekatan *bag of word, vector space, stemming dan stop word* yang bertujuan untuk menentukan *fitur* terbaik dalam mencirikan dokumen.

3. Feature Selection (Attribute Selection)

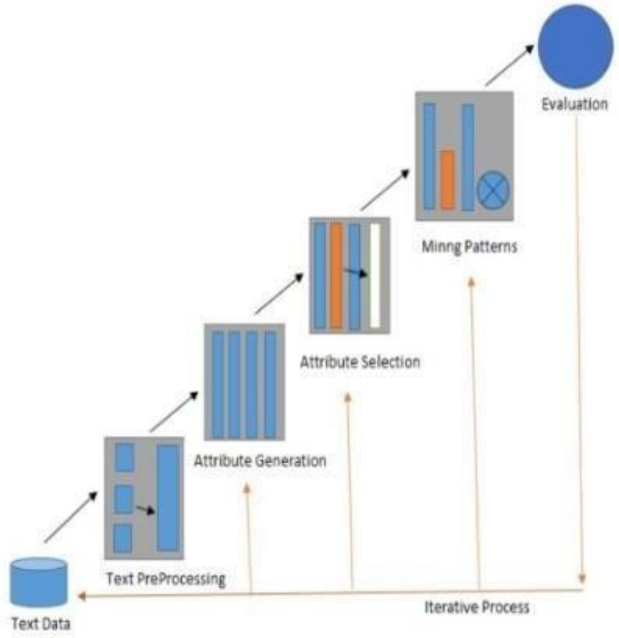
Feature Selection merupakan proses dalam mengurangi *fitur* yang berlebihan atau tidak *relevan* dalam *text* sehingga didapatkan bagian terpenting.

4. Mining Patterns

Mining Pattern merupakan proses *mining* data *text* dengan menggunakan teknik *mining* seperti *classification, clustering dan decision tree* dalam menemukan sebuah pola atau *knowledge* dari *text.*

5. Evaluation

Evaluate melakukan evaluasi dari hasilnya yang selanjutnya dari hasil tersebut dapat dibuang atau dapat digunakan sebagai alur selanjutnya.



Gambar 2.1 Tahapan *Text Mining*

1.4. Sentiment Analisis

Sentiment Analisis atau biasa disebut *opinion mining* merupakan salahsatu cabang penelitian *Text Mining*. *Opinion mining* adalah riset komputasional dari opini, *sentimen* dan emosi yang diekspresikan secara tekstual. Analisis sentimen atau penambangan opini adalah suatu bidang studi untuk menganalisis pendapat orang terhadap entitas seperti produk, layanan,organisasi, individu, masalah, peristiwa, dan topik. Analisis sentimen ini berfokus pada pendapat seseorang yang mengekspresikan atau menyiratkan sentimen positif atau negatif, kebanyakan analis sentimen ini berkaitan dengan orang-orang di media sosial Selain polaritas positif dan negatif, terkadang polaritas juga dianggap sebagai kisaran dimana suatu dokumen dapat berisi penyataan yang memiliki polaritas campuran Mejova, (2009).

Jika diberikan suatu set dokumen teks yang berisi opini mengenai suatu objek, maka *opinion mining* bertujuan untuk mengekstrak *atribut* dan komponen dari objek yang telah dikomentasi pada setiap dokumen dan untukmenentukan apakah komentar tersebut bermakna positif atau negatif. Sentiment Analysis dapat dibedakan berdasarkan sumber datanya, beberapa level yang sering digunakan dalam penelitian Sentiment Analysis, pada level dokumen dan Sentiment Analysis pada level kalimat. Menurut Mejova, (2009) sentiment analysis dapat dipahami pada beberapa level yaitu level dokumen, paragraf, kalimat, atau klausa. Berdasarkan level sumber datanya Sentiment Analysis terbagimenjadi 2 kelompok besar yaitu :

1. ***Coarse-grained* Sentiment Analysis**

   Pada Sentiment Analysis jenis ini, Sentiment Analysis yang dilakukan adalah pada level dokumen. Secara garis besar fokus utama dari Sentiment Analysis jenis ini adalah menganggap seluruh isi dokumen sebagai sebuah sentiment positif atau sentiment negatif.

2. ***Fined-grained* Sentiment Analysis**

   Fined-grained Sentiment Analysis adalah Sentiment Analysis pada level kalimat. Fokus utama *fined-greined* Sentiment Analysis adalah menentukan *sentimen* pada setiap kalimat

1.5. Feature Extraction

Menurut Tu, (2019) *Feature extraction* merupakan langkah awal dalam text mining untuk menemukan *entitas* dan hubungan antar kata dalam suatu dokumen.

Dalam klasifikasi teks, *feature* yang biasa digunakan adalah term, term dapat berupa kata atau frasa yang terdapat dalam dokumen tersebut. *Feature extraction* adalah suatu pengambilan *feature* dari suatu bentuk yang nantinya nilai yang di dapatkan akan dianalisis untuk proses selanjutnya. Hubungan antar kata disini menunjukkan fakta atau peristiwa yang melibatkan *entitas* tertentu. Ada empat tipe elemen yang dapat diambil dari suatu dokumen teks, yaitu:

1. Entitas

   Entitas adalah sesuatu yang memiliki keberadaan yang unik dan berbeda, walaupun tidak harus dalam bentuk fisik. Dalam pengembangan system, entitas digunakan sebagai model yang menggambarkan komunikasi dan pemrosesan internal seperti misalnya membedakan dokumen dengan pemrosesan lainnya.

2. Atribut

   Atribut adalah properti atau karakteristik yang dimiliki oleh suatu entitas. Atribut juga dapat diartikan sebagai suatu fitur hasil ekstraksi dari suatu entitas. Sebagai contoh entitas “Buku” memiliki atribut yaitu nama, pengarang, penerbit, dan lain-lain.

3. Fakta

   Fakta adalah *text mining* yang diartikan sebagai hubungan antar entitas. Sebagai contoh adalah hubungan kerja antara pegawai dan perusahaanyaitu pegawai bekerja untuk perusahaan dan perusahaan mempekerjakan pegawai.

4. Peristiwa

Peristiwa atau *event* adalah suatu kegiatan atau kejadian dimana entitas terlibat didalamnya. Sebagai contoh adalah kegiatan belajar-mengajar dimana entitas “dosen” dan “mahasiswa” terlibat didalamnya.

2.5.1. *Term Frequency Inverse Document Frequency* (TF-IDF)

TFIDF adalah teknik pengambilan informasi yang membebani *term frequency* dan dokumen inversnya (IDF). Setiap kata atau istilah memiliki skor TF dan IDF bobot dari istilah disebut sebagai TFIDF. Algoritma TFIDF digunakan untuk menimbang *keyword* dalam setiap dokumen dan menghitung seberapa banyak kemunculannya dalam setiap dokumen Feldman et al., (2009).

$$IDF = log(\frac{d}{df}), W_{d.t} = tf_{d.t} \cdot IDF.t$$

Dimana:

d: dokumen ke-d

t: kata ke-t dari kata kunci

W: bobot dokumen ke-d terhadap kata ke-t

D: Total dokumen

df: banyak dokumen yang mengandung kata yang dicari

tf: banyak kata yang dicari pada sebuah dokumen

1.6. Algoritma *Support Vector Machine* (SVM)

Permodelan data empiris dapat menimbulkan beberapa permasalahan ketika data yang diperoleh berdimensi tinggi (ruang fitur) dan tidak seragam yang bisa mengakibatkan analisis dengan pendekatan *Neural Network* (NN) tradisional

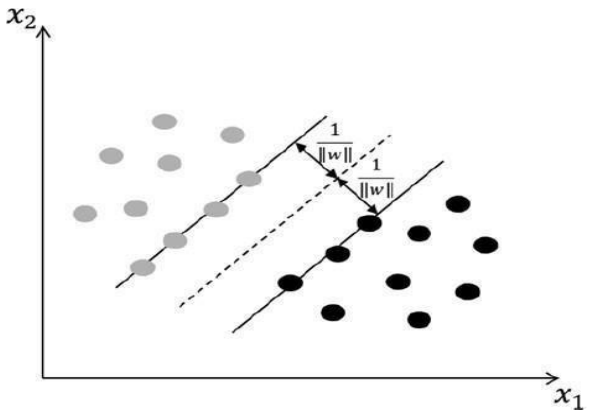
mengalami kesulitan dalam generalisasi dan menghasilkan model yang bisa *overfit* data (Gunn, 1998). SVM dikembangkan untuk memecahkan masalah klasifikasi karena SVM memiliki kemampuan yang lebih baik dalam menggeneralisasi data bila dibandingkan dengan teknik yang sudah ada sebelumnya (Vapnik, Golowich, & Smola, 1997).

SVM merupakan sistem pembelajaran menggunakan ruang berupa fungsi - fungsi linear dalam sebuah ruang fitur yang berdimensi tinggi yang dilatih menggunakan algoritma pembelajaran berdasarkan pada teori optimasi dengan mengimplementasikan *learning bias* (Santosa, 2007). Pendekatan dengan

menggunakan SVM ini memiliki banyak manfaat lain seperti misalnya model yang dibangun memiliki ketergantungan eksplisit pada subset dari datapoints, serta *support vector* yang membantu dalam interpretasi model. Prinsip utama penggunakan SVM adalah mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada ruang input. *Hyperplane* tersebut dapat berupa *line* pada *two dimension* dan dapat berupa *flat plane* pada *multiple plane.* SVM merupakan salah satu *machine learning* yang melakukan pelatihan dengan menggunakan *training dataset* dan melakukan generalisasi dan membuat prediksi dari data baru.

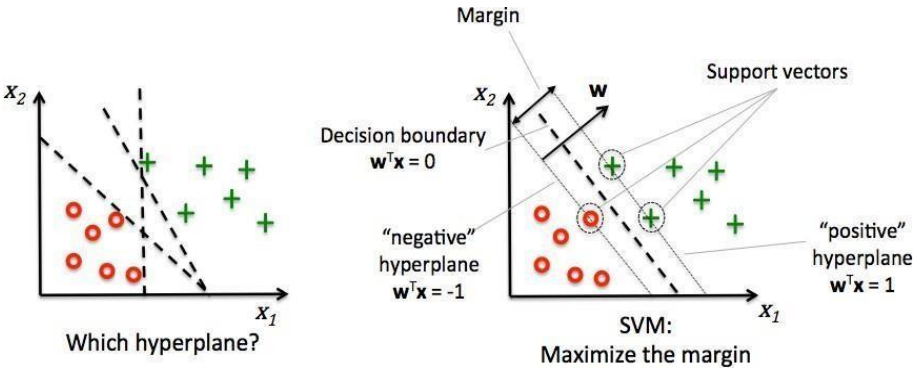
2.6.1. *Hard – Margin* SVM / Linear SVM

Teknik SVM merupakan klasifier yang menemukan *hyperplane* dengan kasus data yang digunakan merupakan data dengan dua kelas yang sudah terpisahsecara linear seperti pada gambar berikut



Gambar 2.2 *Hard margin* SVM

Berdasarkan pada Gambar 2.2 diatas, terlihat bahwa antara kelas positif dan kelas negatif sudah terpisah secara total terlihat dari lingkaran abu – abu yang berada dekat dengan garis x_2 sedangkan untuk lingkaran hitam terletak dekat dengan garis x_1 (Awad & Khanna, 2015).



(Sumber : www.quora.com)

Gambar 2.3 *Hyperplane* terbaik yang memisahkan antar dua kelas positif (+1)dan negatif (-1)

Berdasarkan Gambar 3.2 terlihat beberapa pola yang merupakan anggota dari dua buah kelas yaitu positif (+1) dan negatif (-1). *Hyperplane terbaik* dapat ditemukan dengan mengukur *margin hyperplane* dan mencari titik

maksimalnya. Margin merupakan jarak antara *hyperplane* dengan data terdekat dari masing – masing kelas. Subset *training dataset* yang paling dekat dinamakan sebagai *support vector*. Pada Gambar 3.2 sebelah kanan menunjukkan *hyperplane terbaik*, yaitu terletak pada garis putus – putus yang berada tepat ditengah – tengah *hyperplane* positif dan *hyperplane* negatif. Sedangkan tanda "positif" dan "bulat" yang berada dalam lingkaran hitam merupakan *support vector* (Faihah, 2010).

Pencarian lokasi *hyperplane* optimal merupakan inti dari metode SVM. Diasumsikan bahwa terdapat data *learning* dengan *data points* x_i (i=1,2,...,m) memiliki dua kelas $y_i = \pm 1$ yaitu kelas positif (+1) dan kelas negatif (-1) sehingga akan diperoleh *decision function* berikut.

$f(x) = \text{sign}\ (w.\ x + b)$

Dimana (.) merupakan skalar sehingga $w.x \equiv w^T x$

Berdasarkan pada *decision function* diatas, dapat terlihat bahwa data akan terklasifikasi secara tepat jika $y_i(w.\ x_i + b) > 0$ ✶ᵢ karena ketika $(w.\ x_i + b)$ harus bernilai positif saat $y_i = +1$, dan bernilai negatif ketika $y_i = -1$. *Decision function* menjadi invarian ketika akan dilakukan pembuatan skala positif baru dari argumen dalam persamaan fungsi sehingga akan mengakibatkan ambiguitas dalam mendefinisikan konsep jarak atau margin. Maka dari itu didefinisikan skala untuk (w,b) dengan menetapkan $w.\ x + b =$

1 untuk titik terdekat pada satu sisi dan $w.\ x + b = -1$ untuk titik terdekat

pada sisi lainnya. *Hyperplane* yang melewati $w.\ x\ + b = 1$ dan $w.\ x\ + b\ = -1$ disebut sebagai *hyperplane* kanonik dan wilayah antar *hyperplane* disebut sebagai *margin band* (Cambell & Ying, 2011).

Margin maksimum dapat diperoleh dengan cara memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya.yaitu $\frac{1}{||w||}$ Hal tersebut dirumuskan sebagai *Quadratic Programming* (QP) *Problem* dengan mencari titik minimal seperti pada persamaan berikut.

$\min r(w)=\frac{1}{2}||w||$

Sedangkan subjek *constrain/*kendala persamaannya adalah sebagai berikut

$y_i(w.\ x_i + b) \geq 1$

Persamaan diatas merupakan permasalahan optimisasi kendala dimana kita meminimalkan fungsi objek pada persamaan (3.2) dengan kendala pada persamaan (3.3). Permasalahan diatas dapat direduksi dengan menggunakan fungsi *Lagrange* yang terdiri dari jumlahan fungsi objektif dan m kendala dikalikan dengan pengganda *Lagrange* seperti berikut (Cambell & Ying, 2011).

$(w, b) = \frac{1}{2}(w.w) - \sum_{i=1}^{m} ai(yi(w.xi\ +\ b)\ -\ 1)$

Dimana α_i merupakan *Lagrange Multipliers,* dan nilai $\alpha_i \geq 0$. Pada saat minimum, akan dilakukan penurunan dari *b* dan *w* dan mengaturnya menjadi nolseperti berikut.

$\frac{6L}{6b}$ = - $\sum_{i=1}^{m}$ ai yi = 0

$\frac{6L}{6w}$ = w - $\sum_{i=1}^{m}$ ai yixi = 0

Substitusi nilai *w* dari persamaan kedalam bentuk L(w,b) sehingga akandiperoleh rumus ganda atau biasa disebut sebagai *wolfe dual.*

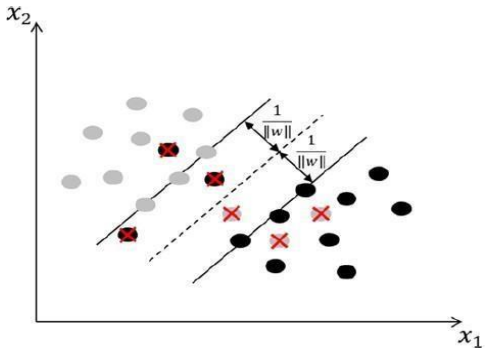
$W(\alpha) = \sum_{i=1}^{m} ai - \frac{1}{2} \sum_{i,j=1}^{m}$ aiajyiyj(xi. xj)

Dimana nilai α_i terhadap kendala adalah sebagai berikut.

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

2.6.2. Soft – Margin SVM

Ketika data yang digunakan tidak sepenuhnya dapat dipisahkan, *slack variables* x_i diperkenalkan kedalam fungsi obyektif SVM untuk memungkinkan kesalahan dalam misklasifikasi. Dalam hal ini, SVM bukan lagi *hard margin classifier* yang akan mengklasifikasi semua data dengan sempurna melainkan sebaliknya yaitu SVM *soft margin classifier* dengan mengklasifikasikan sebagian besar data dengan benar, sementara memungkinkan model untuk membuat misklasifikasi beberapa titik di sekitar batas pemisah. Berikut merupakan gambar ketika data termasuk kedalam *soft margin* SVM (Awad & Khanna, 2015).



Gambar 2.4 Beberapa misklasifikasi pada *Soft margin* SVM

Berdasarkan pada Gambar 3.3 diatas, terlihat bahwa data pada kedua kelastidak terpisah secara sempurna dapat dilihat dari beberapa lingkaran abu – abu yang persebarannya berada di sekitar area lingkaran hitam serta sebaliknyaterdapat beberapa lingkaran hitam yang persebarannya berada di sekitar lingkaran abu – abu. Persamaan *soft margin* hampir miripdengan *hard margin* hanya terdapat sedikit modifikasi dengan adanya *slack variabel* pada persamaan(3.3) sebelumnya seperti berikut

$y_i(w.x_i + b) \geq 1 - s_i$

Kemudian ketika akan dilakukan minimasi jumlahan eror $\sum_{i=1}^{m} \xi 1$

Adalah sebagai berikut.

$min\ [\frac{1}{2} w\,.\,w + C \sum_{i=1}^{m} si\ ] - \sum_{i=1}^{m} ai[yi(w.xi + b) - 1 + si] - \sum_{i=1}^{m} si$

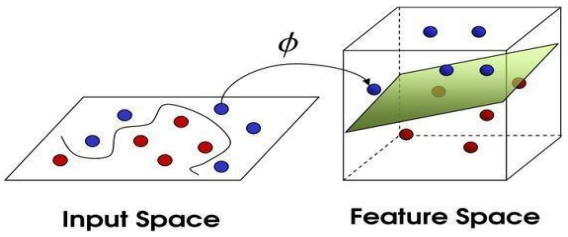
Dengan demikian, persamaan akan diubah kedalam persamaanberikut.

$\min r(w) = \frac{1}{2} |\ w\ |_2 + C \sum_{i=1}^{m} si$

Parameter c digunakan untuk mengontrol *teade off* antara margin dan kesalahan klasifikasi s (Abtohi, 2017).

2.6.3. Kernel SVM

Ketika terdapat permasalahan data yang tidak terpisah secara linear dalam ruang input, *soft margin* SVM tidak dapat menemukan *hyperplane* pemisah yang kuat yang meminimalkan misklasifikasi dari *data points* serta menggeneralisasi dengan baik. Untuk itu, kernel dapat digunakan untuk mentransformasi data ke ruang berdimensi lebih tinggi yang disebut sebagai ruang kernel, dimana akan menjadikan data terpisah secara linear (Awad & Khanna, 2015).



(Sumber : www.quora.com)

Gambar 2.5 Kernel SVM untuk memisahkan data secara linear

Data disimpan dalam bentuk kernel yang mengukur kesamaan atau ketidaksamaan objek data. Kernel dapat dibangun untuk berbagai objek data mulai dari data kontinu dan data diskrit melalui urutan data dan grafik. Konsep substitusi kernel berlaku bagi metode lain dalam analisis data. tetapi SVMmerupakan yang paling terkenal dari metode dengan jangkauan kelas luas yang menggunakan kernel untuk merepresentasikan data dan dapat disebut sebagai metode berbasis kernel (Cambell & Ying, 2011). Berikut merupakan ilustrasi contoh

dalam melakukan pemisahan data menggunakan kernel. Diketahui bahwa data terdiri dari *input space* dengan dua buah x = $\{x_1,x_2\}$ dan z =$\{z_1,z_2\}$. Diasumsikan fungsi kernel akan dibuat dengan menggunakan input x dan z seperti berikut (Rai, 2011).

$K(x, z) = (x^T z)^2$

$K(x, z) = (x_1 z_1 + x_2 z_2)^2$

$K(x, z) = (x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2)^2$

$K(x, z) = \Phi(x)^T \Phi(z)$

Nilai K diatas secara implisit mendefinisikan pemetaan ke ruang dimensi yang lebih tinggi seperti berikut.

$\Phi(x) = \{x_1^2, \sqrt{2}x_1 x_2, x_2^2\}$

Kernel K(x, z) mengambil dua *input space* dan memberikan kesamaannya dalam *feature space* seperti berikut.

$\Phi : X \rightarrow F$

$: X \; x \; X \rightarrow R, K(x, z) = \Phi(x).\Phi(z)$

Berdasarkan pada fungsi kernel diatas, dapat dilakukan perhitungan untuk melakukan prediksi dari beberapa data dalam *feature space* seperti pada persamaan berikut (Cambell & Ying, 2011).

$f(\Phi(x)) = \text{sign}(w.\Phi(z) + b)$

$f(\Phi(x)) = \text{sign}(\sum_{i=1}^{m} a_i y_i K(x, z) + b)$

dimana :

b : nilai bias

m : jumlah *support vector*

K(x, z) : fungsi kernel

Nilai k yang bisa digunakan sebagai fungsi kernel harus memenuhi kondisiMercer antara lain (Rai, 2011) :

- Merupakan *Hilbert Space* dimana nilai *feature space* harus merupakanvektor dengan *dot product.*
- Harus benar jika k merupakan fungsi definit positif

$\int dx \int dz \mathbf{f}(x)\mathrm{K}(\mathrm{x},\mathrm{z})\mathbf{f}(z) > 0 \ (\forall \mathbf{f} \in L_2)$

- Ketika k_1 dan k_2 merupakan fungsi kernel, maka :

  $\mathrm{K}(\mathrm{x}, \mathrm{z}) = \mathrm{K}_1(\mathrm{x}, \mathrm{z}) + \mathrm{K}_2(\mathrm{x}, \mathrm{z})$: *Direct sum*

  $\mathrm{K}(\mathrm{x}, \mathrm{z}) = a\mathrm{K}_1(\mathrm{x}, \mathrm{z})$: *Skalar Product*

  $\mathrm{K}(\mathrm{x}, \mathrm{z}) = \mathrm{K}_1(\mathrm{x}, \mathrm{z})\mathrm{K}_2(\mathrm{x}, \mathrm{z})$: *Direct product*

Berikut merupakan fungsi kernel yang populer dan sering digunakan antara lain sebagai berikut.

1. *Linear* Kernel SVM

Linear kernel merupakan fungsi kernel yang paling sederhana. Linear kernel digunakan ketika data yang dianalisis sudah terpisah secara linear. Linear kernel cocok ketika terdapat banyak fitur dikarenakan pemetaan ke ruang dimensi yang lebih tinggi tidak benar – benar meningkatkan kinerja seperti padaklasifikasi teks. Dalam klasifikasi teks, baik jumlah *instances* (dokumen) maupun jumlah fitur (kata) sama sama besar (Kowalczyk, 2014). Berikut merupakan persamaan dari linear kernel SVM.

$K(x, z) = x^T z$

Pemetaan fungsi Φ merupakan identitas/tidak ada pemetaan

2. *Polynomial* Kernel (derajad d)

Polinomial kernel merupakan fungsi kernel yang digunakan ketika data tidak terpisah secara linear. Polinomial kernel sangat cocok untuk permasalahan dimana semua *training dataset* dinormalisasi.

$K(x, z) = (x^T z)^d$ atau $(1 + x^T z)^d$

3. *Radial Basis Function (RBF)* Kernel

RBF kernel merupakan fungsi kernel yang biasa digunakan dalam analisis ketika data tidak terpisah secara linear. RBF kernel memiliki dua parameter yaitu *Gamma* dan *Cost*. Parameter *Cost* atau biasa disebut sebagai C merupakan parameter yang bekerja sebagai pengoptimalanSVM untuk menghindari misklasifikasi di setiap sampel dalam *training dataset.* Parameter *Gamma* menentukan seberapa jauh pengaruh dari satu sampel *training dataset* dengan nilai rendah berarti "jauh", dan nilaitinggi berarti "dekat". Dengan *gamma* yangrendah, titik yang berada jauh dari garis pemisah yang masuk akal dipertimbangkan dalam perhitungan untuk garis pemisah. Ketika *gamma* tinggi berarti titik – titik berada di sekitar garis yang masuk akal akan dipertimbangkan dalam perhitungan (Patel, 2017). Berikut merupakan persamaan dari RBF kernel.

$K(x, z) = \exp[-y\|x - z\|^2]$

1.7. Optimal

Optimal adalah terbaik, paling menguntungkan (Kamus Besar Bahasa Indonesia Balai Pustaka, 2007). Algoritma yang optimal dapat diartikan urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dalam langkah-langkah terbatas, yang paling baik dan menguntungkan, yang menggambarkan prosedur komputasi tertentu untuk mencapai yang hubungan input / output.

1.8. Algoritma Optimasi

Algoritma Optimasi (Optimization Algorithms) dapat didefenisikan sebagai algoritma untuk menemukan nilai x sedemikian sehingga menghasilkan f(x) yang bernilai sekecil atau sebesar mungkin untuk suatu fungsi f yang diberikan, yang mungkin disertai dengan beberapa batasan pada x. Dimana x bisa berupa skalar atau vektor dari nilai-nilai kontinu maupun diskrit.

Global Optimization didefenisikan sebagai suatu cabang ilmu dari matematika terapan dan analisa numerik yang membahas optimasi dengan kriteria yang bersifat tunggal, ganda atau bahkan mungkin konflik. Kriteria diekspresikan sebagai himpunan fungsi matematika F = {f1, f2,…, fn} yang disebut fungsi- fungsi obbjektif. (Thomas Weise, 2007)

Algoitma optimasi sedikit berbeda dengan algoritma pencarian (search algorithmn). Pada algoritma pencarian terdapat suatu kriteria tertentu yang menyatakan apakah elemen xi merupakan solusi atau bukan. Sebaliknya pada

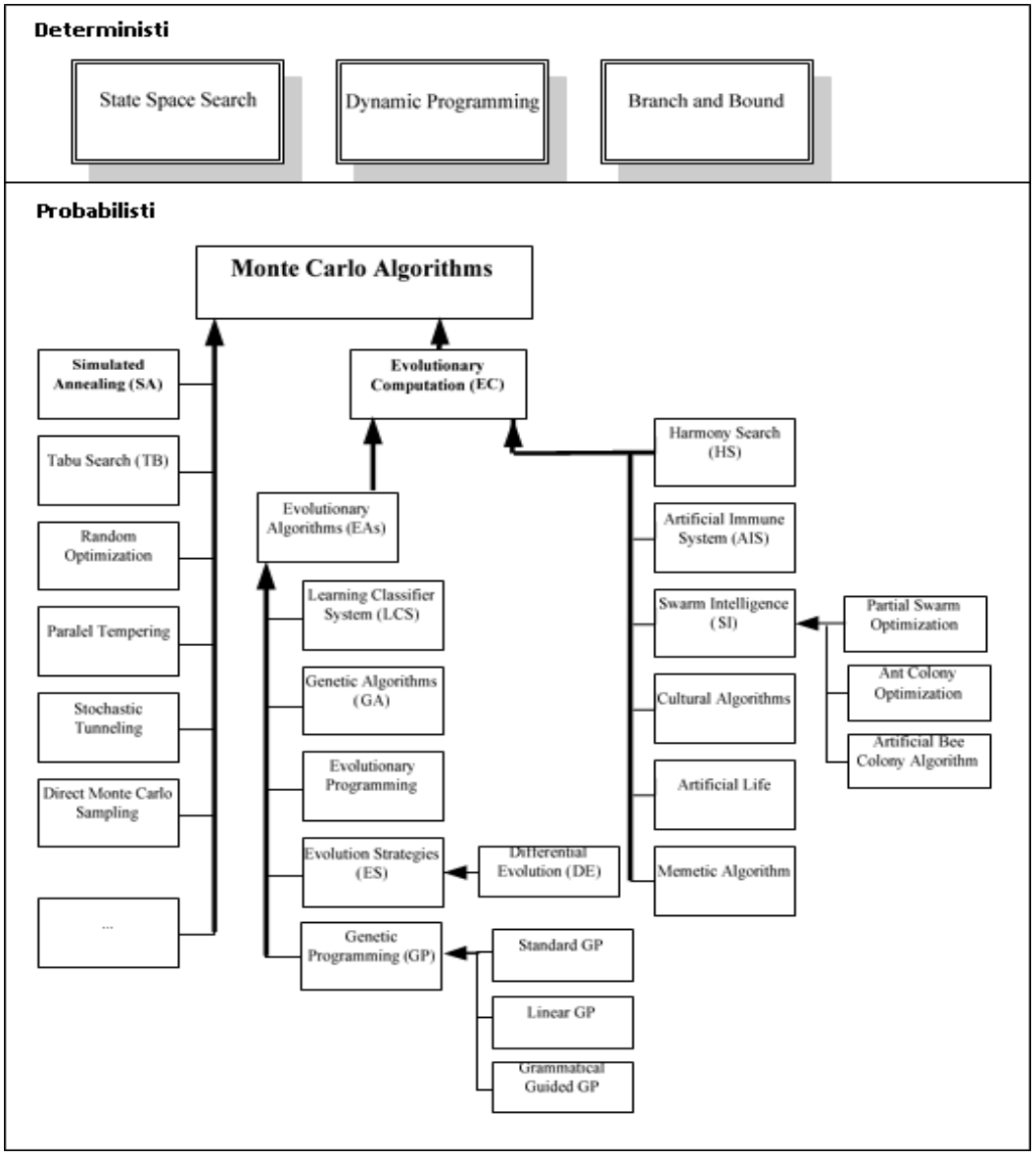
Algoritma optimasi mungkin tidak terdapat kriteria tersebut, melainkan hanya fungsi-fungsi objektif yang menggambarkan bagus tidaknya suatu konfigurasi yang diberikan. Algoritma optimasi bisa dikatakan sebagai generalisasi dari algoritma pencarian atau dengan kata lain, algoritma pencarian adalah kasus khusus dari algoritma optimasi.

1.9. Klasifikasi Algoritma Optimasi

2.9.1. Berdasarkan Metode Operasinya

Berdasarkan metode operasinya, algoritma optimasi dapat dibagi ke dalam dua kelas besar, yaitu algoritma deterministik (deterministic) dan probabilistik (probabilistic). Pada setiap langkah eksekusi dalam algoritma deterministik, terdapat maksimum satu jalan untuk diproses. Jika tidak ada jalan, berarti algoritma sudah selesai. Algoritma deterministik sering digunakan untuk masalah yang memiliki relasi yang jelas antara karakteristik calon solusi dengan utilitasnya. Dengan demikian, ruang pencarian dapat dieksplorasi menggunakan, misalnya, metode branch and bound atau divide and conquer scheme. Tetapi, jika relasi antara suatu kan-didat solusi dan "fitness"-nya tidak dapat dipahami atau diamati (kandidat-kandidat solusi yang bertetangga mungkin sangat berbeda dalam hal utilitas atau dimensi ruang pencarian yang sangat besar), maka masalah ini akan lebih sulit diselesaikan secara deterministik. Bayangkan jika kita harus mencari suatu solusi dari 101000000 kandidat solusi secara deterministik. Tentu membutuhkan usaha dan waktu yang banyak. Hal ini samasulitnya dengan mencari jarum di tumpukan jerami. Demikianlah perbedaan signifikan antara algoritma deterministik dan algoritma probabilistik. Untuk permasalahan tertentu, kita bisa menggunakan algoritma deterministik. Tetapi, untuk masalah yang lain mungkin sajakita harus menggunakan algoritma probabilistik. Bagaimanapun tidak

ada satupun algoritma optimasi yang sesuai untuk semua masalah. Biasanya suatu algoritma sangat efektif dan efisien untuk beberapa masalah, tetapi algoritma tersebut mungkin saja memiliki performansi yang kurang atau bahkan sangat buruk untuk permasalahan yang lain.



Gambar 2.6 Klasifikasi Algoritma berdasarkan metode operasinya.

2.9.2. Berdasarkan akurasi dan kecepatan

Pengklasifikasian algoritma optimasi di atas secara sekilas, sudah cukup memadai jika dipandang secara teori. Perbedaan prinsip dasar dan diikuti algoritma sangat jelas terlihat. Cara klasifikasi lain yang bisa cligunakan adalah membedakan algoritma optimasi menjadi dua, yaitu: optimasi online dan optimasi offline.

A. Optimasi Online

Sesuai dengan namanya, optimasi ini ditujukan untuk permasalahan yang membutuhkan solusi dalam waktu cepat dan biasanya permasalahan tersebut terjadi secara berulang-ulang. Misalnya, penentuan lokasi robot (robot localization), penyeimbangan beban (load balancing), komposisi layanan untuk proses bisnis dari IT system yang sedang berjalan di suatu perusahaan, atau perbaruan jadwal pekerjaan mesin (machine job schedule) di suatu pabrik setelah datangnya suatu permintaan baru.

B. Optimasi Offline

Optimasi jenis ini ditujukan untuk permasalahan yang membutuhkan solusi tidak dalam waktu cepat dan biasanya masalah ini terjadi dalam periode waktu yang lama. Misalnya, optimasi dalam proses perancangan (design optimization), data mining, pembuatan jadwal kuliah setiap semester, dan sebagainya. User bisa memiliki waktu lama, hingga berhari-hari, untuk menunggu proses optimasi menghasilkan solusi yang seoptimal

mungkin (kalau bisa optimum global). Untuk masalah seperti ini dan ruang masalahnya yang tidak terlalu besar, algoritma optimasi yang bersifat deterministik biasanya menghasilkan solusi yang lebih baik dibandingkan optimasi probabilistic.

2.9.3. Berdasarkan Analog dan Nama

Bardasarkan analogidan nama yang digunakan, algoritma optimasi menjadi dua : algoritma maksimasi dan algoritma minimasi.

A. Algoritma Maksimasi

Algoritma ini menggunakan analogi memaksimalkan sesuatu pada dunia nyata. Salah satu contohnya adalah algoritma Hill Climbing (HC), menggunakan analogi pendekatan bukit untuk mencapai puncak tertinggi

B. Algoritma Minimasi

Algoritma ini menggunakan analogi meminimalkan sesuatu pada dunia nyata. Salah satu contohnya adalah algoritma Simulated Annealing (SA) yang mensimulasikan proses annealing pada pembuatan materi yang terdiri dari butir kristal atau logam. Tujuan dari proses ini adalah menghasilkan struktur kristal yang baik dengan menggunakan energi seminimal mungkin. Salah satu studi kasus yang menggunakan Simulated Annealing yaitu job shop scheduling, yang akan dijelaskan lebih lanjut pada bagian pembahasan dari makalah ini.

1.10. Algoritma *Particle Swarm Optimization* (PSO)

Particle Swarm Optimization (PSO) adalah teknik optimasi yang dikembangkan oleh Kennedy dan Eberhart pada tahun 1995, populasi dari PSO berdasarkan strategi optimisasi stokastik. PSO terinspirasi oleh perilaku sosial kawanan burung, ikan (Hapid hapani, 2020).

Particle Swarm Optimization (PSO) untuk memecahkan masalah optimasi global dalam bentuk algoritma *metaheuristik* paralel. Dalam beberapa tahun terakhir, banyak algoritma *metaheuristik* telah maju. PSO adalah salah satu dari mereka, sangat efektif untuk memecahkan masalah ini. Tapi PSO memiliki beberapa kekurangan seperti *konvergensi prematur* dan terjebak dalam minimum lokal

Algoritma *Particle Swarm Optimization* (PSO) adalah teknik optimasi berdasarkan populasi yang terinspirasi oleh perilaku sosial dari pergerakan burung atau ikan (*bird flocking* atau *fish schooling*). PSO sebagai alat optimasi menyediakan prosedur pencarian berbasis populasi dimana masing-masingindividu yang disebut partikel mengubah posisi mereka terhadap waktu (Rosita, Yudhi, dan Rully, 2012). Pada sistem PSO, masing-masing partikel terbang mengitari ruang pencarian multi dimensional (*multidimensional search space*) danmenyesuaikan posisinya berdasarkan pengalaman pribadinya dan pengalaman partikel di sebelahnya.

Berikut adalah langkah-langkah dalam *Particle Swarm Optimization*:

1. Inisialiasi: Inisialisasi kecepatan awal pada iterasi ke-0, dapat dipastikan bahwa nilai kecepatan awal semua partikel adalah 0. Inisialisasi posisi

awal partikel pada iterasi ke-0, posisi awal partikel dibangkitkan dengan persamaan .

$$x = xmin + [0,1] \times (xmax - xmin)$$

Keterangan :

x = posisi partikel

$xmin$ = posisi partikel minimal

$xmin$ = posisi partikel maximal

$rand[0,1]$= nilai random antara 0 dan 1 berdistribusi uniform dalam interval 0 dan 1.

Inisialisasi pBest dan gBest pada iterasi ke-0, pBest akan disamakan dengan nilai posisi awal partikel. Sedangkan gBest dipilih dari satu pBest dengan fitness tertinggi.

2. Perbaharui Kecepatan: Perbaharui kecepatan dengan rumus :

$$v_{i,j}^{t+1} = w.v_{i,j}^{t} + c_1.r_1\left(pBest_{i,j}^{t} - x_{i,j}^{t}\right) + c_2.r_2\left(gBest_{g,j}^{t} - x_{i,j}^{t}\right)$$

Keterangan :

$v_{i,j}^{t+1}$ = kecepatan partikel i dimensi j pada iterasi t

w = bobot inersia

$v_{i,j}^{t}$ = kecepatan partikel i dimensi j pada iterasi t

w = bobot inersia

$c1$ = konstanta kecepatan 1

$c1$ = konstanta kecepatan 2

$r1$, $r2$ = nilai acak antara 0 dan 1

$pBest_{i,j}^{t}$ = posisi terbaik dari partikel i dimensi j pada iterasi t

$gBest_{g,j}^{t}$= global optimum dari partikel g dimensi j pada iterasi t

$x_{i,j}^{t}$ = posisi partikel i dimensi j

3. Perbaharui Posisi dan Hitung Fitness: Perbaharui posisi dengan rumus :

$$x_{i,j}^{t+1} = x_{i,i}^{t} + v_{i,j}^{t+1}$$

Keterangan :

$x_{i,j}^{t+1}$ = posisi partikel i dimensi j pada iterasi t

$x_{i,j}^{t}$ = posisi partikel i dimensi j

$v_{i,j}^{t+1}$ = kecepatan partikel i dimensi j pada iterasi t dan hitung fitness dengan nilai akurasi dari model terbaik.

4. Perbaharui *pBest* dan *gBest*: Dilakukan perbandingan antara *pBest* pada iterasi sebelumnya dengan hasil dari update posisi. *Fitness* yang lebih tinggi akan menjadi *pBest* yang baru. *pBest* terbaru yang memiliki nilai *fitness* tertinggi akan menjadi *gBest* yang baru.

1.11. Twitter

Twitter merupakan sosial media yang dikembangkan oleh seseorang bernama *Jack Dorsey* pada tahun 2006, *Twitter* adalah *social networking* dimana seseorang bisa berkomunikasi dengan yang lainnya melalui layanan pesan singkat bernama *tweets*, setiap *tweet* nya pengguna dibatasi menggunakan 140 karakter.

Sehingga pengguna bisa lebih singkat dalam membuat *tweet* Gragg & Sellers, (2010).

Application Programming Interface (*API*) merupakan fungsi-fungsi/perintah-perintah untuk menggantikan bahasa yang digunakan dalam system calls dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh programmer. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil system calls sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari system callssama dengan nama di API.

Twitter API merupakan opsi pengembang yang ditawarkan oleh *Twitter*, *Twitter* memiliki dasar API *(Application Programming Interface)* dimana setiap pengguna dapat mengembangkan program yang bisa tergabung dengan layanan *Twitter*, *Twitter* API mengizinkan pengguna mengakses data Gragg & Sellers, (2010).

Pada awalnya perusahaan Summize yang menyediakan fasilitas mencari data di Twitter. Kemudian perusahaan Summize ini diakuisisi dan diganti merek menjadi TwitterSearch sehingga Search API terpisah sebagai entitas sendiri. Twitter API terdiri dari 3 (tiga) bagian yaitu:

A. *Search API.*

   Search API dirancang untuk memudahkan *user* dalam mengelola *query search* di konten *Twitter*. *User* dapat menggunakannya untuk mencari*tweet* berdasarkan *keyword* khusus atau mencari *tweet* lebih spesifik berdasarkan *username* Twitter. *Search API* juga menyediakan akses pada data *Trending* Topik.

B. *Representational State Transfer (REST) API*

REST API memperbolehkan *developer* untuk mengakses inti dari *Twitter* seperti *timeline*, status *update* dan informasi *user*. *REST API* digunakan dalam membangun sebuah aplikasi *Twitter* yang kompleks yang memerlukan inti dari *Twitter*.

C. *Streaming API*

Streaming API digunakan developer untuk kebutuhan yang lebih intensif seperti melakukan penelitian dan analisis data. *Streaming API* dapat menghasilkan aplikasi yang dapat mengetahui statistik status update, *follower* dan lain sebagainya.

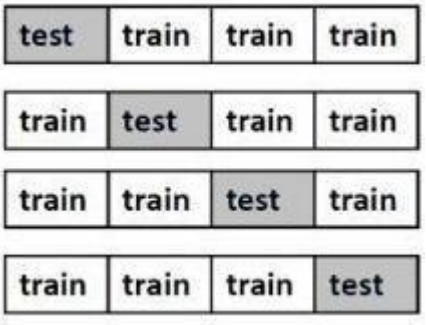
Dalam penelitian ini, bagian *Twitter API* yang digunakan adalah *REST API.*

1.12. K-Fold Cross Validation

Menurut Rohani, Abbas., et al. (dalam Jiang, Ping., 2017) *K-Fold Cross Validation* merupakan salah satu bagian dari jenis pengujian *cross validation* yang berguna untuk menilai kinerja proses sebuah metode algoritma dengan membagi sampel data secara *random* dan mengelompokkan data tersebut sebanyak nilai K k-fold. selanjutnya salah satu kelompok k-fold tersebut akan dijadikan sebagai data uji sedangkan sisa kelompok yang lain akan dijadikan sebagai data *training*.

Data yang digunakan dibagi secara acak ke dalam k subset yaitu $D1$,$D2$, … ,Dk dengan ukuran yang sama. Dataset akan dibagi menjadi data training dan data testing. Proses training dan testing dilakukan sebanyak k kali secara

berulang-ulang. Pada iterasi ke-i, partisi Di disajikan sebagai data testing dan partisi sisanya digunakan secara bersamaan dan berurutan sebagai data training. Iterasi kedua, subset $D1$,$D2$, … ,Dk akan dites pada $D2$, dan selanjutnya hingga Dk (Han, et al, 2012: 364). Gambar 2.5 berikut adalah contoh ilustrasi 4-fold cross validation.



Gambar 2.6. Ilustrasi 4-Fold Cross Validation

Berdasarkan Gambar 2.5 ditunjukkan bahwa nilai fold yang digunakan adalah 4-fold cross validation. Berikut diberikan langkah-langkah pengujian data dengan 4-fold cross validation.

a. Dataset yang digunakan dibagi menjadi 4 bagian, yaitu $D1$, $D2$,3, dan $D4$. , t = (1, 2, 3, 4) digunakan sebagai data testing dan dataset lainnya sebagai data training.
b. Tingkat akurasi dihitung pada setiap iterasi (iterasi-1, iterasi-2, iterasi-3, iterasi-4), kemudian dihitung rata-rata tingkat akurasi dari seluruh iterasi untuk mendapatkan tingkat akurasi data keseluruhan

1.13. Bahasa Pemograman Python

Python merupakan bahasa pemrograman yang diinterpretasikan secara umum, interaktif, berorientasi objek, dan bahasa pemorgraman tingkat tinggi. Bahasa ini diciptakan oleh Guido Van Rossum selama tahun 1985-1990. Seperti *Perl, source code python* tersedia dibawah GNU *General Public License* (GPL) Menczer et al., (2020).