MODUL DATA MINING

REVIEW MATERI
PERTEMUAN 13 (ONLINE)

Disusun Oleh
Syefira Salsabila

Kata *Mining* merupakan kiasan dari bahasa inggris, mine. Jika mine berarti menambang sumber daya yang tersembunyi di dalam tanah, maka Data Mining merupakan penggalian makna yang tersembunyi dari kumpulan data yang sangat besar. Karena itu *Data Mining s*ebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), m*achine learning*, statistik dan basis Data.

Data Mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu:

1. Classification
   Suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Salah satu contoh yang mudah dan popular adalah dengan Decision tree yaitu salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasi. Decision tree adalah model prediksi menggunakan struktur pohon atau struktur berhirarki.

2. Association
   Digunakan untuk mengenali kelakuan dari kejadian-kejadian khusus atau proses dimana hubungan asosiasi muncul pada setiap kejadian. Salah satu contohnya adalah Market Basket Analysis, yaitu salah sati metode asosiasi yang menganalisa kemungkinan pelanggan untuk membeli beberapa item secara bersamaan.

3. Clustering
   Digunakan untuk menganalisis pengelompokkan berbeda terhadap data, mirip dengan klasifikasi, namun pengelompokkan belum didefinisikan sebelum dijalankannya tool data mining. Biasanya menggunkan metode *neural network* atau statistik. Clustering membagi item menjadi kelompok-kelompok berdasarkan yang ditemukan tool data mining.

Metode pelatihan yaitu cara berlangsungnya pembelajaran dan pelatihan dalam menganalisis dataset untuk mendapatkan pola data tertentu. Metode pelatihan data mining memiliki 3 kelompok, seperti : *supervised learning*, *unsupervised learning*, dan *association learning*. 3 kelompok tersebut memiliki definisi, sebagai berikut.

1. *Supervised Learning*
   Kumpulan record dari inputan yang digunakan dan telah diketahui output , dengan kata lain variable yang menjadi target telah ditentukan dalam *dataset* yang sedang dianalisis. Sebagian besar algoritma dalam kelompok tersebut terdiri dari : klasifikasi, estimasi, dan prediksi. Algoritma yang digunakan akan melakukan *process* pembelajaran yang berdasarkan *value* dari *variable* sasaran yang telah terasosiasi dengan *value* pada varibel *predictor*.

2. *Unsupervised Learning*
   Pada metode tersebut data yang dianalisa diterapkan tanpa adanya guru serta pelatihan pada data lampau, dengan kata lain diartikan sebagai pencarian pola pada setiap atribut yang digunakan. Tidak termasuk penetapan atribut atau kelas pada sasaran. Contoh algoritma yang menerapkan metode *unsupervised learning* adalah Clustering.
3. *Association Learning*
   Berbeda dengan dua kelompok yang terdapat di atas, pada mode ini mempunyai tujuan untuk mencari atribut yang muncul pada transaksi yang sama. Algoritma asosiasi biasanya berfungsi untuk mencari dan menganalisa transaksi belanja dengan konsep mencari produk yang dibeli secara bersamaan dalam satu transaksi yang sama. Algoritma yang digunakan dalam kelompok asosiasi adalah Apriori.

Dalam dunia data mining atau data science sering kali kita mendengar *supervised* dan *unsupervised learning.* Secara garis besar terdapat 2 pendekatan untuk melakukan teknik - teknik data mining. ***Supervised learning*** **adalah sebuah pendekatan dimana sudah terdapat data yang dilatih, dan terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengkelompokan suatu data ke data yang sudah ada,** lain halnya dengan unsupervised learning, ***unsupervised learning*** **tidak memiliki data latih, sehingga dari data yang ada, kita mengelompokan data tersebut menjadi 2 bagian atau 3 bagian dan seterusnya.**

S*upervised learning* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Sedangkan pada *unsupervised learning*, data belum memiliki pola apapun, dan tujuan *unsupervised learning* untuk menemukan pola dalam sebuah data. Contoh Supervised Learning adalah ketika Anda memiliki sejumlah buku yang sudah dilabeli dengan kategori tertentu. Misalnya, kategori buku novel seperti Digital Fortress, Inferno, Deception Point. Kategori buku akademik, seperti Pengantar Teknologi Informasi, R in Action, Rekayasa Perangkat Lunak. Kategori biografi antara lain Anne Frank, Abraham Lincoln dan Mandela. Selanjutnya, ketika Anda membeli sejumlah buku baru, maka Anda harus mengindentifikasi isi dari buku tersebut, dan memasukannya dalam kategori. Ketika Anda membeli buku Logika fuzzy, Anda pasti akan memasukan buku tersebut ke dalam buku akademik.

Lain halnya dengan *Unsupervised Learning.* Anda tidak memiliki data yang dilatih sebelumnya. Anggaplah Anda belum pernah membeli buku sama sekali, namun dalam satu hari, Anda membeli banyak tumpukan buku dan ingin membaginya kedalam beberapa kategori agar nantinya mudah dicari. Anda akan mengidentifikasi buku buku mana yang mirip. Dalam hal ini, kita memilih pendekatan buku berdasarkan isinya. Misalnya anda memiliki buku Digital Fortress, Inferno, Deception Point, Pengantar Teknologi Informasi, R in Action, Rekayasa Perangkat Lunak, Anne Frank, Abraham Lincoln dan Mandela. Anda akan mengklasifikasikan buku Pengantar Teknologi

Informasi, R in Action, Rekayasa Perangkat Lunak Anda ke dalam buku akademik karena keperluannya untuk kuliah.

Untuk melakukan hal itu Anda perlu algoritma yang mendukung untuk pengimplementasian dari metode tersebut.

Algoritma Supervised Learning:

a. Decision tree
b. Nearest - Neighbor Classifier
c. Naive Bayes Classifier
d. Artificial Neural Network
e. Support Vector Machine
f. Fuzzy K-Nearest Neighbor

Algoritma Unsupervised Learning

a. K-Means
b. Hierarchical Clustering
c. DBSCAN
d. Fuzzy C-Means
e. Self-Organizing Map

Kesimpulannya dari penjelasan di atas adalah jika anda memiliki data data sebelumnya dan memiliki variabel target yang akan diklasifikasikan, maka Anda dapat memakai metode *supervised learning.* Jika Anda ingin membagi data - data tersebut ke dalam beberapa kelompok maka Anda memakai metode u*nsupervised learning*

A. Decision Tree

Seperti diketahui bahwa manusia selalu menghadapi berbagai macam masalah di dalam kehidupannya sehari-hari. Masalah-masalah yang timbul dari berbagai macam bidang ini memiliki tingkat kesulitan dan kompleksitas yang sangat bervariasi, mulai dari masalah yang sangat sederhana dengan sedikit faktor-faktor terkait hingga masalah yang sangat rumit dengan banyak sekali faktor-faktor yang terkait, sehingga factor-faktor yang berkaitan dengan masalah tersebut perlu untuk diperhitungkan.

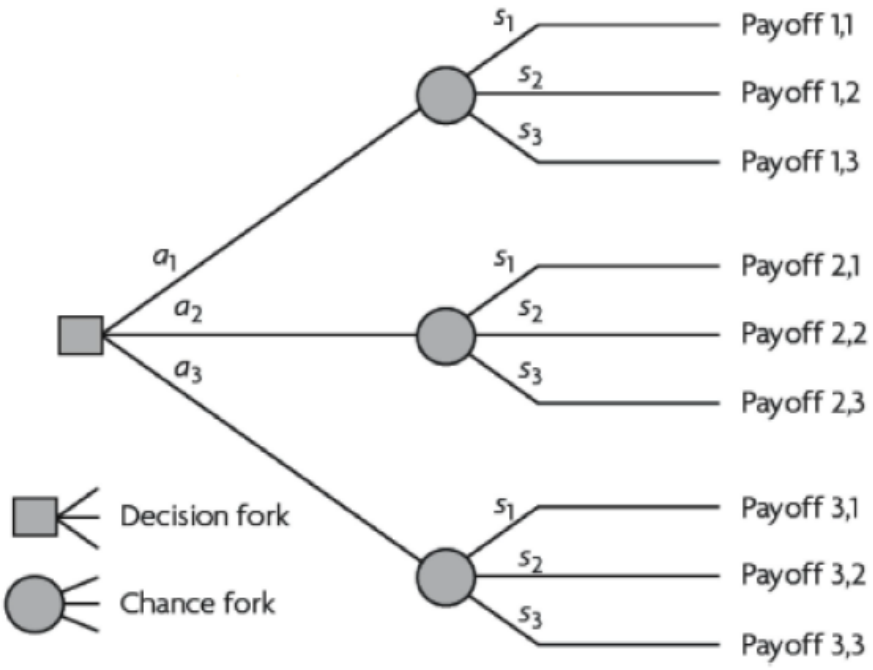
Seiring dengan perkembangan kemajuan pola pikir manusia, manusia mulai mengembangkan sebuah sistem yang dapat membantu manusia dalam menghadapi masalah-masalah yang timbul sehingga dapat menyelesaikannya dengan mudah. Pohon keputusan atau yang lebih dikenal dengan istilah *Decision Tree* ini merupakan implementasi dari sebuah sistem yang manusia kembangkan dalam mencari dan membuat keputusan untuk masalah-masalah tersebut dengan memperhitungkan berbagai macam faktor yang berkaitan di dalam lingkup masalah tersebut.

Dengan pohon keputusan, manusia dapat dengan mudah mengidentifikasi dan melihat hubungan antara faktor-faktor yang mempengaruhi suatu masalah sehingga dengan memperhitungkan faktor-faktor tersebut dapat dihasilkan penyelesaian terbaik untuk masalah tersebut. Pohon keputusan ini juga dapat menganalisa nilai risiko dan nilai suatu informasi yang terdapat dalam suatu alternatif pemecahan masalah.

Pohon keputusan dalam analisis pemecahan masalah pengambilan keputusan merupakan pemetaan alternatif-alternatif pemecahan masalah yang dapat diambil dari masalah tersebut. Pohon keputusan juga memperlihatkan faktor-faktor kemungkinan yang dapat mempengaruhi alternative-alternatif keputusan tersebut, disertai dengan estimasi hasil akhir yang akan didapat bila kita mengambil alternatif keputusan tersebut.

Secara umum, pohon keputusan adalah suatu gambaran permodelan dari suatu persoalan yang terdiri dari serangkaian keputusan yang mengarah kepada solusi yang dihasilkan. Peranan pohon keputusan sebagai alat bantu dalam mengambil keputusan telah dikembangkan oleh manusia sejak perkembangan teori pohon yang dilandaskan pada teori graf. Seiring dengan perkembangannya, pohon keputusan kini telah banyak dimanfaatkan oleh manusia dalam berbagai macam sistem pengambilan keputusan.

Decision tree adalah struktur flowchart yang menyerupai tree (pohon), dimana setiap simpul internal menandakan suatu tes pada atribut, setiap cabang merepresentasikan hasil tes, dan simpul daun merepresentasikan kelas atau distribusi kelas. Alur pada decision tree di telusuri dari simpul akar ke simpul daun yang memegang prediksi.



Gambar 4.1 Bentuk Decision Tree Secara Umum

a) Algoritma c4.5

Pohon keputusan merupakan metode yang umum digunakan untuk melakukan klasifikasi pada data mining. Seperti yang telah dijelaskan sebelumnya, klasifikasi merupakan Suatu teknik menemukan kumpulan pola atau fungsi yang mendeskripsikan serta memisahkan kelas data yang satu dengan yang lainnya untuk menyatakan objek tersebut masuk pada kategori tertentu dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan.

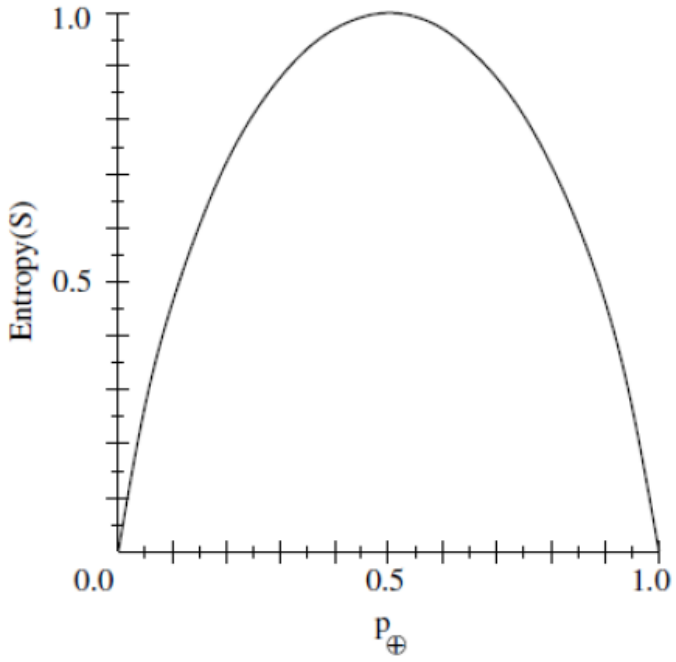
Metode ini popular karena mampu melakukan klasifikasi sekaligus menunjukkan hubungan antar atribut. Banyak algoritma yang dapat digunakan untuk membangun suatu decision tree, salah satunya ialah algoritma C45.

Algoritma C4.5 dapat menangani data numerik dan diskret. Algoritma C.45 menggunakan rasio perolehan (gain ratio). Sebelum menghitung rasio perolehan, perlu dilakukan perhitungan nilai informasi dalam satuan bits dari suatu kumpulan objek, yaitu dengan menggunakan konsep entropi.

Konsep Entropy

Entropy(S) merupakan jumlah bit yang diperkirakan dibutuhkan untuk dapat mengekstrak suatu kelas (+ atau -) dari sejumlah data acak pada ruang sampel S.

Entropy dapat dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. semakin kecil nilai Entropy maka akan semakin Entropy digunakan dalam mengekstrak suatu kelas. Entropi digunakan untuk mengukur ketidakaslian S.



Gambar 4.2 Grafik Entropi

Besarnya Entropy pada ruang sampel S didefinisikan dengan:

$$\boldsymbol{Entropy(S) \equiv -p_{\oplus}\, log_2\, p_{\oplus} - p_{\ominus}\, log_2 p_{\ominus}}$$

Dimana:

- S : ruang (data) sampel yang digunakan untuk pelatihan
- $p_{\oplus}$: jumlah yang bersolusi positif atau mendukung pada data sampel untuk kriteria tertentu
- $p_{\ominus}$: jumlah yang bersolusi negatif atau tidak mendukung pada data sampel untuk kriteria tertentu.

- Entropi(S) = 0, jika semua contoh pada S berada dalam kelas yang sama.
- Entropi(S) = 1, jika jumlah contoh positif dan negative dalam S adalah sama.
- 0 > Entropi(S) > 1, jika jumlah contoh positif dan negative dalam S tidak sama.

Konsep Gain

Gain (S,A) merupakan Perolehan informasi dari atribut A relative terhadap output data S. Perolehan informasi didapat dari output data atau variabel dependent S yang dikelompokkan berdasarkan atribut A, dinotasikan dengan gain (S,A).

$$Gain(S,A) \equiv Entropy(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy(Si)$$

Dimana:

- A : Atribut
- S : Sampel
- n : Jumlah partisis himpunan atribut A
- $|S_i|$: Jumlah sampel pada pertisi ke –i
- $|S|$: Jumlah sampel dalam S

Untuk memudahkan penjelasan mengenai algoritma C4.5berikut ini disertakan contoh kasus yang dituangkan dalam Tabel 4.1:

Tabel 4.1 Keputusan Bermain Tenis

No	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Cloudy	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	Yes
7	Cloudy	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Cloudy	Mild	High	TRUE	Yes
13	Cloudy	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

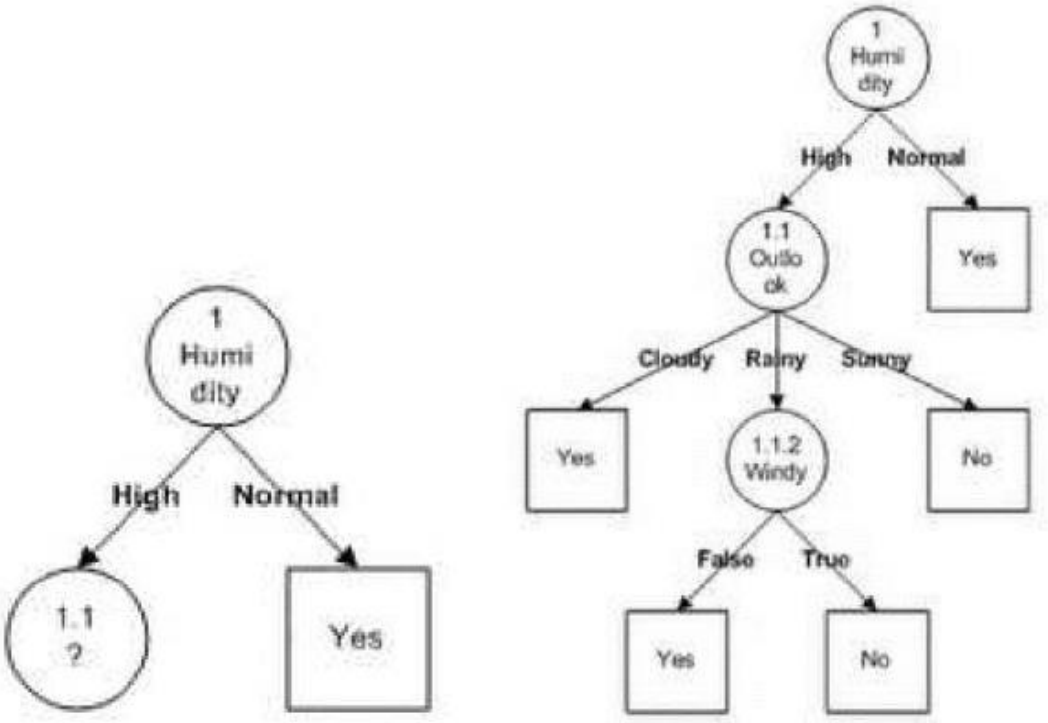
Tabel 1 merupakan kasus yang akan dibuat pohon keputusan untuk menentukan main tenis atau tida. Data ini memiliki atribut-atribut yaitu, keadaan cuaca (outlook), temperatur, kelembaban (humidity) dan keadaan angin (windy).

Berikut merupakan cara membangun pohon keputusan dengan menggunakan algoritma:

1. Pilih atribut sebagai akar. Sebuah akar didapat dari nilai gain tertinggi dari atribut-atribut yang ada.
2. Buat cabang untuk masing-masing nilai
3. Bagi kasus dalam cabang
4. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Tabel 4.2 Perhitungan Simpul 1

NODE			JUMLAH KASUS	NO (S_1)	YES (S_2)	ENTROPY	GAIN
1	TOTAL		14	4	10	0.863120569	
	OUTLOOK						0.258521037
		CLOUDY	4	0	4	0	
		RAINY	5	1	4	0.721928095	
		SUNNY	5	3	2	0.970950594	
	TEMPERATURE						0.183850925
		COOL	4	0	4	0	
		HOT	4	2	2	1	
		MILD	6	2	4	0.918295834	
	HUMIDITY						0.370506501
		HIGH	7	4	3	0.985228136	
		NORMAL	7	0	7	0	
	WINDY						0.005977711
		FALSE	8	2	6	0.811278124	
		TRUE	6	4	2	0.918295834	



Dari hasil pada Tabel 4.2 dapat diketahui bahwa atribut dengan Gain tertinggi adalah HUMIDITY yaitu sebesar 0.37. Dengan demikian HUMIDITY dapat menjadi node akar. Ada 2 nilai atribut dari HUMIDITY yaitu HIGH dan NORMAL. Dari kedua nilai atribut tersebut, nilai atribut NORMAL sudah mengklasifikasikan kasus menjadi 1 yaitu keputusan-nya Yes, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut HIGH masih perlu dilakukan perhitungan lagi hingga semua kasus masuk dalam kelas seperti yang terlihat pada Gambar di sebelah kanan.

Kelebihan Pohon Keputusan

Dalam membuat keputusan dengan menggunakan pohon keputusan, metode ini memiliki kelebihan sebagai berikut:

a. Daerah pengambilan keputusan lebih simpel dan spesifik.
b. Eliminasi perhitungan-perhitungan tidak diperlukan, karena ketika menggunakan metode pohon keputusan maka sample diuji hanya berdasarkan kriteria atau kelas tertentu.
c. Fleksibel untuk memilih fitur dari internal node yang berbeda. Sehingga dapat meningkatkan kualitas keputusan yang dihasilkan jika dibandingkan ketika menggunakan metode penghitungan satu tahap yang lebih konvensional.
d. Dengan menggunakan pohon keputusan, penguji tidak perlu melakukan estimasi pada distribusi dimensi tinggi ataupun parameter tertentu dari

distribusi kelas tersebut. Karena metode ini menggunakan kriteria yang jumlahnya lebih sedikit pada setiap node internal tanpa banyak mengurangi kualitas keputusan yang dihasilkan.

Kekurangan Pohon Keputusan

Pohon keputusan sangat membantu dalam pengambilan keputusan, namun pohon keputusan juga memiliki beberapa kekurangan, diantaranya:

a. Kesulitan dalam mendesain pohon keputusan yang optimal.
b. Hasil kualitas keputusan yang didapat sangat tergantung pada bagaimana pohon tersebut didesain. Sehingga jika pohon keputusan yang dibuat kurang optimal, maka akan berpengaruh pada kualitas dari keputusan yang didapat.
c. Terjadi overlap terutama ketika kelas-kelas dan criteria yang digunakan jumlahnya sangat banyak sehingga dapat menyebabkan meningkatnya waktu pengambilan keputusan dan jumlah memori yang diperlukan.
d. Pengakumulasian jumlah error dari setiap tingkat dalam sebuah pohon keputusan yang besar.

Isu Terkait Decision Tree

Sekali decision tree dibangun berdasarkan objek yang dimiliki dalam data latih, maka pohon tersebut sebenarnya alami dan sedikit atau banyak sudah merefleksikan objek-objek tersebut. Biasanya banyak cabang yang secara kuat dipengaruhi anomali data (data yang menyimpang) yang mungkin ada di set data. Data seperti ini disebut noise atau outlier. Data-data yang menyimpang seperti ini sebaiknya sudah dilakukan pemangkasan di awal pemrosesan sehingga tidak mempengaruhi kinerja algoritme utama yang digunakan dalam data mining. Secara prinsip, jika pohon dibangun dari data mentah yang belum mengalami pemrosesan awal sama sekali, maka dipastikan bahwa decision tree secara penuh merefleksikan semua isi set data latih. Karena decision tree dibangun untuk menyelesaikan kasus klasifikasi hingga sisa terkecil, maka decision tree bisa mengalami keadaan yang di atas normal. Di sisi lain, jika decision tree yang dibangun terlalu simpel terhadap data latih yang digunakan untuk membangunnya maka akan mengalami keadaan yang di bawah normal terhadap data latih. Untuk menangani masalah tersebut, maka diperlukan adanya pemrosesan pemangkasan (pruning) cabang yang memberikan informasi redundan berulang), atau yang tidak mengikuti pola data umumnya. Dengan cara ini, maka akan didapatkan pohon yang tidak terlalu ‘rindang; tetapi lebih besar skalabilitas dan kecepatan prediksinya.

Ada dua jenis pemangkasan dalam decision tree:

a. Pre-pruning
   Pendekatan ini berarti bahwa secara praktik akan menghentikan ‘pertumbuhan’ selama proses induksi pohon dengan memilih berhenti pada sebuah node, yang kemudian node tersebut akan menjadi daun dan diberikan label kelas sesuai dengan elemen data terbanyak. Syarat utama penggunaan pendekatan

ini adalah bahwa semua objek data dimiliki oleh kelas yang sama atau semua nilai fiturnya sama

b. Post-pruning
   Pendekatan ini digunakan setelah pohon tumbuh lengkap. Pendekatan 'bottom-up' didasarkan pada nilai error prediksi. Node akan dipangkas dengan membuang cabang. Akibatnya, node menjadi daun dan diberi label kelas sesuai dengan elemen data terbanyak. Error prediksi dapat dikurangi dengan cara ini.

B. Naïve Bayes

Naïve Bayes adalah teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema Bayes (aturan Bayes) dengan sebuah asumsi independensi (ketidaktergantungan) yang kuat (naif). Dapat dikatakan, pada *Naïve Bayes* model yang digunakan adalah "model fitur independen". Dalam Bayes (terutama *Naïve Bayes*), makna independensi yang kuat pada fitur adalah bahwa sebuah fitur dalam suatu data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama.

Teori keputusan *bayes* adalah pendekatan statistik yang fundamental dalam pengenalan pola (*pattern recognition)* , pendekatan ini didasarakan pada kuantifikasi *trade-off* antara berbagai keputusan klasifikasi dengan menggunakan probabilitas dan ongkos yang di timbulkan dalam keputusan tersebut. Selain itu *Bayesian clasification* juga dapat memprediksi probabilitas keanggotaan suatu *class*. pada teorema *bayes* yang memiliki kemampuan klasifikasi serupa dengan *decision tree* dan *neural network* . *Bayesian classification* terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam *database* dengan data yang besar.

Algoritma *naïve* Bayes merupakan salah satu algoritma yang terdapat pada teknik klasifikasi. *Naive* Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukan oleh ilmuwan Inggris *Thomas* Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai teorema Bayes. Teorema tersebut dikombinasikan dengan *naive* di mana diasumsikan kondisi antar atribut saling bebas. Klasifikasi *naive* Bayes diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya.

Naive Bayes digunakan untuk memprediksi peluang dimasa yang akan datang berdasarkan pengalaman pada waktu lampau, sehingga disebut dengan Teorema Bayes. Teorema ini digabungkan dengan Naive yang mengasumsikan kondisi antar elemen saling bebas. *Classification* Naive Bayes diupayakan bahwa ada atau tidak ada ciri tertentu dari sebuah *class* tidak ada hubungannya dengan ciri dari kelas lainnya.

Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan yaitu seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik dari yang dilakukan manusia. Salah satu penerapan yaitu pada sistem pakar (*Expert System).* Sistem pakar adalah suatu sistem yang berbasis komputer yang menggunakan pengetahuan, fakta serta teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut.

Salah satu penerapan dalam sistem pakar ada pada bidang kesehatan yaitu sitem pakar penyakit pada ibu hamil menggunakan pendekatan metode naïve bayes yaitu menetukan penyakit berdasarkan gejala umum yang diderita oleh seorang ibu hamil serta dengan menghitung peluang seorang ibu hamil mengidap penyakit kehamilan dan memberikan memberikan solusi pencegahan berdasarkan pada jenis penyakit yang diderita. Penerapan system pakar yang berguna untuk mendapatkan informasi tentang awal penyakit yang dialami ibu hamil dan apakah system pakar dengan metode *Naïve Bayes* ini dapat digunakan untuk mendaptkan informasi tentang awal penyakit yang terjadi pada masa kehamilan yang diakibatkan oleh gangguan yang muncul akibat kehamilan tersebut.

Metode *Naïve Bayes Classifier* merupakan pengklasifikasi probabilitas yang sederhana yang diberdasarkan pada teorema Bayes. Teorema Bayes dikombinasi dengan "*Naïve*" yang artinya setiap atribut atau variabel bersifat bebas (*independent*). *Naïve Bayes Classifier* bisa dilatih dengan efisien dalam suatu pembelajaran terawasi *(supervised learning)*. Keuntungan klasifikasi ini adalah hanya membutuhkan jumlah kecil data pelatihan yang berguna untuk memperkirakan parameter (sarana dan varians dari variabel) yang diperlukan dalam klasifikasi. Karena variabel independen diasumsikan, hanya variasi dari variabel untuk masing-masing kelas harus ditentukan, bukan seluruh matriks kovarians. *Naive Baye Classifier* tersebut mengestimasi peluang kelas bersyarat dengan mengasumsikan atributnya yaitu independen secara bersyarat yang diberikan label dengan kelas y. Teorema perhitungan *Naive Bayes Classifire* berdasarkan probabilitas sebagai berikut

Persamaan dari teorema Bayes adalah

$$P(C \mid F) = \frac{P(C).P(F \mid C)}{P(F)} \quad (1)$$

Keterangan :

$P(C|F)$:Probabilitas akhir bersyarat (*posterior*) suatu Kelas C terjadi jika diberikan Petunjuk (atribut) F terjadi

$P(C)$:Probabilitas awal (*prior*) Kelas C terjadi tanpa memandang petunjuk (atribut) apapun

$P(F|C)$:Probabilitas sebuah petunjuk (atribut) F terjadi akan mempengaruhi Kelas C

$P(F)$:Probalilitas awal (*prior*) petunjuk (atribut) F terjadi tanpa memandang Kelas apapun

Adapun alur dari metode Naive Bayes, sebagai berikut:

1. Membaca data training
2. Hitung Jumlah dan probabilitas, namun apabila data numerik maka.
   a. Mencari *value* rata-rata dan standar deviasi dari tiap-tiap parameter yang merupakan data berangka. Adapun kesamaan yang menggunakan dalam nilai rata-rata hitung (*mean*) dapat dijabarkan sebagai berikut.

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \quad (2.2)$$

dimana :

μ : rata-rata hitung (*mean*)

X_i: nilai sample ke-i

n : jumlah sample

Dan persamaan dalam hitung *value* simpangan baku (standar deviasi), sebagai berikut.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n-1}} \qquad (2.4)$$

dimana :

σ : standar Deviasi

X_i : nilai x ke-i

μ : rata-rata hitung (*mean*)

n : jumla sample

b. Mencari *value* probabilitas dengan langkah hitung jumlah data yang sesuai dari bilangan yang sama dibagi dengan jumlah data pada kategori tersebut.

3. Memperoleh *value* dalam tabel rata-rata, standar deviasi dan peluang.

Kelebihan *Naive Bayes*:

1. Mudah diimplementasikan
2. Hasilnya *robust* untuk data yang memuat *noisy* dan untuk data yang tidak berkaitan
3. Dapat menangani *missing value*
4. Hasilnya cukup baik untuk sebagian besar kasus

Kekurangan *Naive Bayes*:

1. Adanya asumsi saling bebas antar atributnya terkadang akan menurunkan tingkat akurasinya

2. Biasanya dalam kehidupan nyata selalu ada hubungan antar atribut sehingga asumsi saling bebas menjadi tidak dipenuhi

FUTURE SELECTION

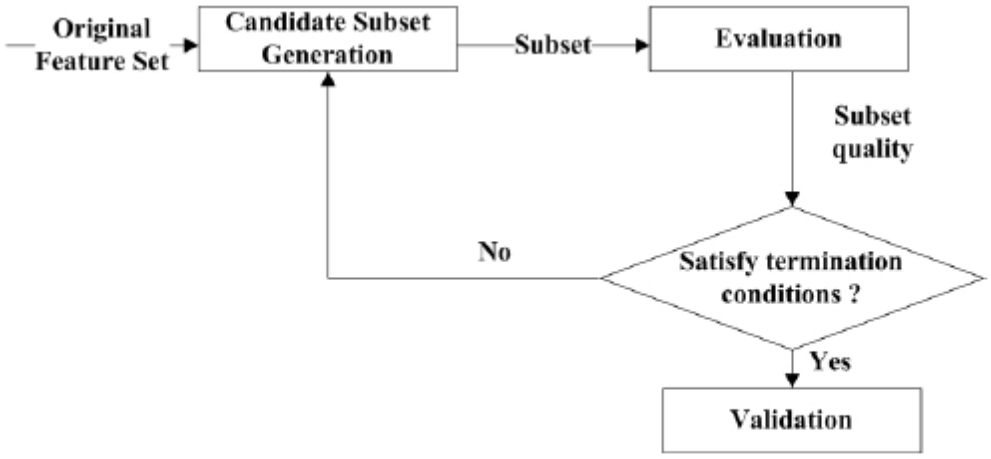
Informasi menjadi salah satu kebutuhan paling dasar manusia dan menjadi komoditi yang penting, dimana saaat ini tak dapat dipungkiri kita sudah berada pada era "*information-based society*". Informasi dapat diartikan suatu data atau objek yang diproses terlebih dahulu sedemikian rupa sehingga dapat tersusun dan terklasifikasi dengan baik, sehingga memiliki arti bagi penerimanya yang selanjutnya menjadi pengetahuan bagi penerima tentang suatu hal tertentu yang membantu pengambilan keputusan secara tepat. Informasi memiliki sifat *integrity, availability* (ketersediaan), dan *confidentiality* (kerahasiaan), dan informasi bagi sebuah perusahaan adalah modal sangat penting. Dari ketiga sifat itu jika ada yang terganggu maka keamanan sistem dan jaringan (*system and network security*) patut diperhatikan dengan seksama dan harus diperbaiki.

Menjadi hal penting yang harus diperhatikan dalam keamanan sistem informasi dan jaringan komputer:

a. Kehilangan data / *data loss*
b. Penyusup / *intruder*

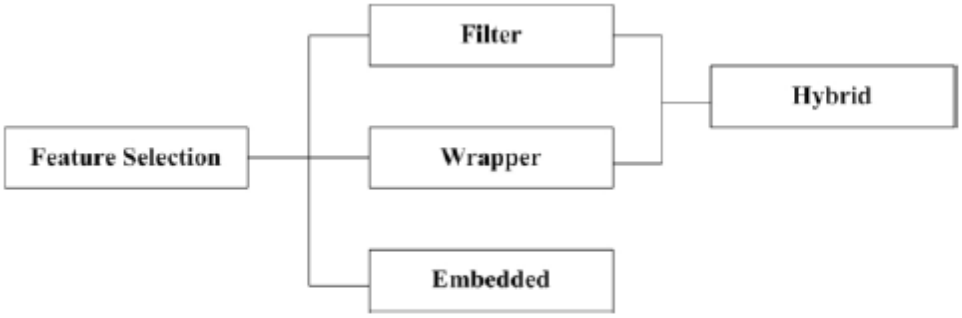
Menurut Bace dan Mell penyusupan/*intrusion* adalah kegiatan yang merusak atau menyalahgunakan sistem atau setiap usaha yang melakukan compromise integritas kepercayaan atau ketersediaan suatu sumber daya komputer dan tidak bergantung pada berhasil atau tidaknya aksi tersebut sehingga ini berkaitan dengan suatu serangan pada sistem komputer.

Seleksi fitur adalah satu dari istilah yang umum digunakan dalam data mining. Digunakan untuk mengurangi input sesuai ukuran yang akan dikelola pada processing dan analisis. Fitur atau atribut pada dataset KDD CUP'99 diselidiki untuk mengindentifikasi relevansi setiap fitur dalam metode induksi. Rule deteksi intrusi digunakan untuk menentukan fitur yang paling diskriminatif untuk masing-masing kelas. Sehingga relevansi dari 41 fitur yang berkaitan dengan label dataset dapat diselidiki.



Gambar 7. Proses seleksi fitur [21]

Ada 4 model utama yang ditetapkan pada seleksi fitur yaitu: metode *wrapper*, metode *filter*, metode *hybrid* dan metode *embedded.*



Gambar 8. 4 metode seleksi fitur [21]

Feature selection adalah suatu metode penganalisaan data yang bertujuan untuk memilih fitur yang berpengaruh (fitur optimal) dan mengesampingkan fitur yang tidak berpengaruh. Ada beberapa algoritma *feature selection* yang dapat digunakan, salah satunya adalah *Relief*. *Relief* memanfaatkan teknik bobot (*weight*) untuk mengukur signifikansi fitur dalam konteks klasifikasi dan fitur yang memiliki nilai bobot di atas ambang batas (*threshold*) yang digunakan akan dipilih.

Suatu objek perlu diketahui fitur-fiturnya agar dapat dikenali dan dibedakan dari objek yang lain. Fitur-fitur optimal yang dapat diketahui dari suatu objek akan mempermudah dan mempercepat proses identifikasi objek tersebut. Fitur atau variabel di dalam penelitian merupakan suatu atribut dari sekelompok objek yang diteliti yang mempunyai variasi antara satu dengan yang lain dalam kelompok tersebut.

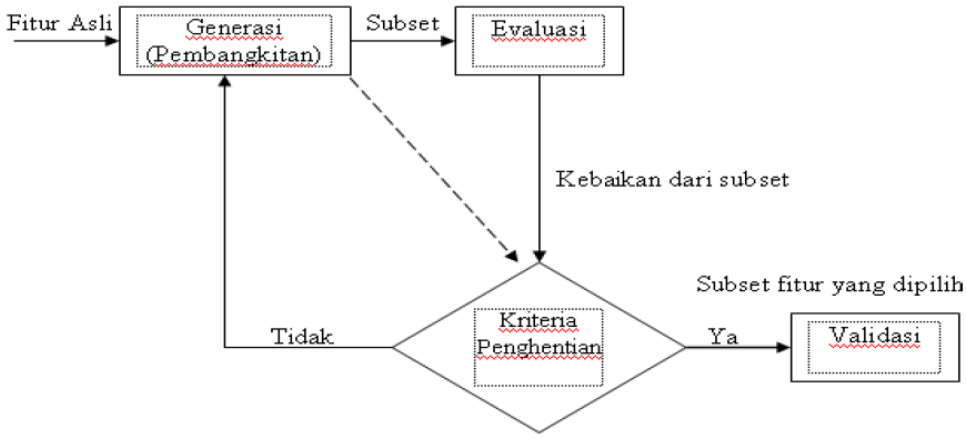
Feature Selection merupakan suatu kegiatan pemodelan atau penganalisaan data yang umumnya dapat dilakukan secara *preprocessing* dan bertujuan untuk memilih fitur yang berpengaruh (fitur optimal) dan mengesampingkan fitur yang tidak berpengaruh. Ada beberapa algoritma *Feature Selection* yang dapat digunakan. untuk menemukan fitur-fitur yang optimal dari sebuah himpunan fitur. Salah satu algoritma *Feature Selection* adalah algoritma *Relief*. *Relief* pertama kali diusulkan oleh Kira dan Rendell pada tahun 1992. *Relief* termasuk dalam metode *Feature Selection* tipe *Filter*, yang didasarkan pada estimasi fitur. *Relief* memberikan nilai yang relevan untuk setiap fitur, dan fitur yang memiliki nilai di atas ambang batas (*threshold)* yang diberikan oleh pengguna yang akan dipilih. Algoritma *Relief* memanfaatkan teknik bobot untuk mengukur signifikansi fitur dalam konteks klasifikasi. Bobot *Relief* adalah nilai-nilai yang kontinu dan memungkinkan fitur untuk digolongkan berdasarkan relevansi. *Relief* juga merupakan algoritma yang menarik dalam *Feature Selection* karena memiliki komputasi yang efisien.

Feature Selection digunakan untuk menemukan subhimpunan dari himpunan fitur yang tersedia untuk meningkatkan aplikasi dari suatu algoritma pembelajaran. *Feature Selection* digunakan dibanyak area aplikasi sebagai alat untuk menghilangkan fitur yang tidak relevan dan atau fitur berlebihan. Sebuah fitur dikatakan tidak relevan jika memberikan sedikit informasi, sedangkan sebuah fitur dikatakan berlebihan jika informasi yang diberikan adalah informasi yang terkandung dalam fitur lain (tidak memberikan informasi baru).

Ada empat langkah yang dilakukan dalam feature selection yaitu:

a. **Prosedur generasi (pembangkitan)**, untuk menghasilkan calon subhimpunan berikutnya dapat dilakukan dengan beberapa cara yaitu : lengkap, heuristik dan acak.
b. **Evaluasi fungsi**, untuk mengevaluasi subhimpunan, dengan cara mengukur jarak, informasi, konsistensi, ketergantungan, dan mengukur tingkat kesalahan klasifikasi.
c. **Kriteria penghentian,** untuk memutuskan kapan harus berhenti, dengan cara melihat nilai ambang batas *(threshold)*, diawali dengan sejumlah pengulangan dan sebuah ukuran subhimpunan fitur terbaik.
d. **Prosedur validasi**, untuk memeriksa apakah subhimpunan valid. (opsional).

Proses dalam feature selection tersebut dapat dituangkan dalam skema berikut:



Gambar 1. Proses *Feature Selection* dengan validasi, (Dash dan Liu, 1997)

Prosedur Generasi

Prosedur generasi merupakan prosedur pencarian yang pada dasarnya menghasilkan *subset* (subhimpunan) dari fitur-fitur untuk dievaluasi. Jika himpunan fitur asli berisi *N* jumlah fitur, maka jumlah calon bersaing untuk menjadi subhimpunan yang dihasilkan adalah *2N*. Ini merupakan jumlah besar bahkan untuk setengah dari jumlah *N*. Ada berbagai pendekatan untuk menyelesaikan masalah ini, yaitu: lengkap, heuristik, dan acak.

a. Lengkap
   Urutan ruang pencarian prosedur generasi ini adalah *O*(*2N*), sebuah subhimpunan yang sedikit untuk dievaluasi. Subhimpunan fitur yang optimal sesuai dengan evaluasi fungsi, karena prosedur ini dapat dilakukan dengan cara mundur. Mundur dapat dilakukan dengan menggunakan berbagai teknik, seperti: *branch and bound*, pencarian pertama terbaik, dan balok pencarian.
b. Heuristik
   Dalam setiap pengulangan prosedur generasi ini, semua sisa fitur yang belum dipilih (ditolak) masih dipertimbangkan untuk pemilihan (penolakan). Ada banyak variasi untuk proses sederhana ini, tapi generasi subhimpunan pada dasarnya meningkat atau menurun. Urutan ruang pencarian adalah *O*(*N2*) atau kurang. Prosedur ini sangat sederhana untuk diterapkan dan sangat cepat dalam memperoleh hasil, karena ruang pencarian hanya kuadrat dari jumlah fitur.
c. Acak
   Prosedur generasi ini masih baru dalam penggunaannya dalam metode *Feature Selection* dibandingkan dengan dua kategori lainnya. Meskipun ruang pencarian adalah *O*(*2N*), tetapi metode ini biasanya mencari lebih sedikit jumlah subhimpunan daripada *2N* dengan menetapkan jumlah maksimum pengulangan yang mungkin. Optimalitas subhimpunan yang

dipilih tergantung pada sumber daya yang tersedia. Setiap prosedur generasi acak akan memerlukan nilai-nilai dari beberapa parameter.

Evaluasi Fungsi

Evaluasi fungsi mengukur kebaikan subhimpunan yang dihasilkan oleh beberapa prosedur generasi, dan nilai ini dibandingkan dengan yang terbaik sebelumnya. Jika ditemukan yang lebih baik, maka subhimpunan terbaik sebelumnya digantikan. Ada beberapa cara dalam melakukan evalusi fungsi, salah satunya yaitu ukuran Jarak**.**

Juga dikenal sebagai keterpisahan, perbedaan, atau diskriminasi ukuran. Untuk dua kelas, fitur *X* adalah fitur yang lebih disukai dari fitur *Y* apabila *X* menginduksi perbedaan yang lebih besar antara kedua kelas probabilitas kondisional dari *Y* dan jika perbedaan adalah nol, maka *X* dan *Y* tidak dapat dibedakan (sama). Sebagai contoh adalah jarak *Euclidea*. *Euclidean* merupakan metode pengukuran jarak di antara dua objek berdasarkan akar jumlah kuadrat jarak kedua objek. Rumus umum untuk menghitung jarak *Euclidean* yaitu, jika *X* memiliki koordinat (x1, x2,…xn) dan objek *Y* memiliki koordinat (y1, y2, …yn), maka jarak *Euclidean* kedua objek tersebut adalah,

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)}$$

Kriteria Penghentian

Prosedur generasi dan evaluasi fungsi dapat mempengaruhi pilihan untuk kriteria penghentian.

Prosedur Validasi

Proses validasi bukan merupakan bagian dari proses *Feature Selection* itu sendiri, namun sebuah *Feature Selection* harus divalidasi dengan cara melakukan pengulangan terhadap evaluasi fungsi subhimpunan dari fitur sampai kriteria penghentian terpenuhi. Dengan terus bertambahnya jumlah dan keanekaragaman dokumen, penggolongan secara manual tentu saja akan menjadi suatu masalah baru untuk user. Hal tersebut akan memakan banyak waktu dan menimbulkan kejenuhan. Dokumen yang tersebar dan tidak terkoordinasi dengan baik akan menyulitkan user dalam mendapatkan informasi yang diinginkan. Dengan dokumen yang telah terklasifikasi, pengguna informasi (*user*) dapat dengan mudah menemukan dokumen yang dibutuhkan karena dokumen tersebut telah dikelompokkan berdasarkan kategori yang mencerminkan isi dari suatu dokumen. Sekumpulan data biasanya diklasifikasikan dalam single label dengan jumlah atribut yang terbatas.

Untuk meningkatkan efisiensi dan keakuratan dalam klasifikasi dokumen diperlukan teknik *feature selection.* Teknik ini mengurangi jumlah fitur yang ada pada

feature space dan juga merupakan salah satu pemecahan dalam menangani data *imbalance*. Berawal dari masalah tersebut, maka dalam tugas akhir ini akan dilakukan analisis perbandingan metode *feature selection* untuk menangani data *imbalance* pada suatu klasifikasi dokumen. Diantaranya adalah metode *Odds Ratio* (OR) dan *GSS Coefficient* yang termasuk kedalam *feature selection* menggunakan *onesided metric* dimana *one-sided metric* hanya memilih fitur positif yang berpengaruh pada kelas. Kemudian *Information Gain* (IG) yang termasuk kedalam *two-sided metrics* dimana *two-sided metric* mengkombinasikan secara implisit fitur positif dan fitur negatif. Selain itu metode *improved OR* (iOR) dan *improved SIG* (iSIG) yang termasuk kedalam kombinasi antara fitur positif dan fitur negatif secara eksplisit.

Pendekatan *feature selection* yang akan dilakukan terdiri atas *filtering feature selection* dan *wrapper feature selection*. Pada penerapan *filtering feature selection,* selain menggunakan metode *multinomial naive bayes* juga akan dilakukan proses pengklasifikasian dokumen menggunakan metode yang. Metode *multinomial naive bayes* merupakan algoritme yang *naive* karena mengasumsikan indepedensi di antara kemunculan kata-kata dalam dokumen, tanpa memperhitungkan urutan kata dan informasi konteks dalam kalimat atau dokumen secara umum. Selain itu metode tersebut memperhitungkan jumlah kemunculan kata dalam dokumen.

Kemajuan teknologi dalam penyimpanan data telah mendorong banyak orang untuk berlomba-lomba menyimpan data mereka, baik berupa data pribadi sampai data perusahaan yang besar sekalipun. Hal ini dapat mengakibatkan jumlah data yang semakin meningkat dari hari ke hari. Hal ini dapat menimbulkan masalah. Apabila data hanya dikumpulkan dan ditumpuk begitu saja, maka data tersebut tidak lebih hanyalah sebuah tumpukan data-data setelah digunakan untuk kepentingan operasional tertentu.

Oleh sebab itu, muncullah apa yang disebut dengan *Data Mining*, yaitu proses menemukan ilmu atau sesuatu yang berguna dari suatu basis data atau kumpulan data yang besar. Pengertian sesuatu yang berguna memiliki kepentingan dan pengertian yang berbeda bagi tiap orang. *Data Mining* memiliki beberapa fungsi atau tugas seperti klasifikasi, klasterisasi, asosiasi, deteksi anomali, dan prediksi.Dalam melaksanakan tugasnya, tentunya setiap fungsi dari *Data Mining* tersebut dihadapkan dengan berbagai tantangan, seperti skalabilitas, dimensionalitas, heterogenitas, *robustness,* kepemilikan dan distribusi data.

Skalabilitas merupakan permasalahan yang berkaitan dengan jumlah data yang sangat besar, dimensionalitas berkaitan dengan jumlah dimensi data yang banyak, heterogenitas mengenai tipe data inputan yang beraneka ragam, *robustness* berkaitan dengan masalah *noise,* data yang tidak lengkap, dll, kepemilikan dan distribusi data berkaitan dengan pendistribusian data berukuran sangat besar sehingga harus dimiliki oleh beberapa basis data maupun media penyimpanan yang berbeda. Beberapa masalah yang dihadapi oleh berbagai fungsi *Data Mining* tersebut banyak berhubungan dengan jumlah data, dimensi data, dan jumlah atribut data yang sangat besar.

Diperlukan suatu mekanisme untuk meningkatkan kinerja fungsi *Data Mining* agar lebih optimal. Fungsi tersebut dibutuhkan sebelum proses *Data Mining* tersebut dijalankan, yaitu proses *preprocessing*. Pada proses *preprocessing* ada banyak cara yang dilakukan agar inputan data yang diterima diolah sedemikian rupa agar fungsi *Data Mining* menjadi lebih optimal, diantaranya: *Feature Selection, Dimension Reduction, Normalization, Data subsetting.* Salah satu langkah yang akan diambil pada Tugas Akhir ini adalah mengenai *Feature Selection. Feature Selection* dapat membantu mengolah data dengan jumlah fitur/atribut yang sangat banyak. Jumlah waktu dan memori yang dibutuhkan oleh *Data Mining* juga akan berkurang dengan pengurangan dimensionalitas.

Feature Selection dapat membantu fungsionalitas dari tugas *Data Mining* seperti Klasifikasi, Klasterisasi, dll. Kerja dari fungsi *Data Mining* akan tidak optimal apabila didalamnya terdapat fitur/atribut yang tidak relevan dan redundan. *Feature Selection* dapat membuang fitur/atribut yang tidak relevan dan yang redundan serta meningkatkan performansi dari tugas *Data Mining*. *Feature Selection* juga banyak dilakukan pada bidang yang lainnya seperti stastika, *image retrieval, pattern recognition,* dll. Metode yang banyak dilakukan pada *Feature Selection* secara umum terbagi menjadi 2, yaitu metode *filter* dan metode *wrapper*. Pada metode *filter* seleksi fitur dilakukan sebelum algoritma *Data Mining* (pada saat ini klasterisasi) dijalankan.

Pendekatan yang dilakukan pada *Feature Selection* ini adalah dengan menggunakan metode *wrapper.* Metode *wrapper* merupakan metode yang menggunakan algoritma dari *Data Mining* target (pada saat ini klasterisasi) sebagai sebuah *black box* untuk menemukan subset atribut terbaik. Ide dasar dari metode ini adalah melakukan pencarian melalui subset ruang fitur dengan input seluruh fitur dari dataset. Lalu evaluasi setiap subset fitur kandidat dengan pertama mengklaster dengan algoritma klasterisasi dan kemudian mengevaluasi klaster hasil subset fitur dengan menggunakan kriteria *Feature Selection* yang terpilih. Proses ini akan diulang hingga ditemukan subset fitur terbaik dengan klaster yang sesuai berdasarkan kriteria evaluasi fitur. Pendekatan *wrapper* membagi tugas kedalam tiga tahap: pencarian fitur, algoritma klasterisasi, dan evaluasi subset fitur. Pendekatan *wrapper* menjadi menarik karena dilakukan dengan cara menggabungkan algoritme klasterisasi pada pencarian dan pemilihan fitur.

Jumlah informasi pada artikel berbahasa Indonesia berbasiskan web saat ini semakin besar. Besarnya jumlah ini menyebabkan diperlukannya suatu kategorisasi terhadap artikel tersebut untuk memudahkan pembaca dalam mencari topik berita yang mereka inginkan. Salah satu cara yang dapat dilakukan sebagai solusi untuk masalah ini adalah dengan menggunakan proses kategorisasi teks dalam *data mining* yang dapat menggali informasi yang tersembunyi dari informasi-informasi mentah yang ada.

Akan tetapi, tingginya dimensi dari *feature space* data dan adanya data *noise* menjadi masalah utama dalam kategorisasi teks. Hal ini dapat mengganggu efektifitas dari hasil kategorisasinya itu sendiri. Oleh karena itu, harus dilakukan pemilihan terhadap beberapa atribut yang dapat berpengaruh besar terhadap hasil kategorisasi,

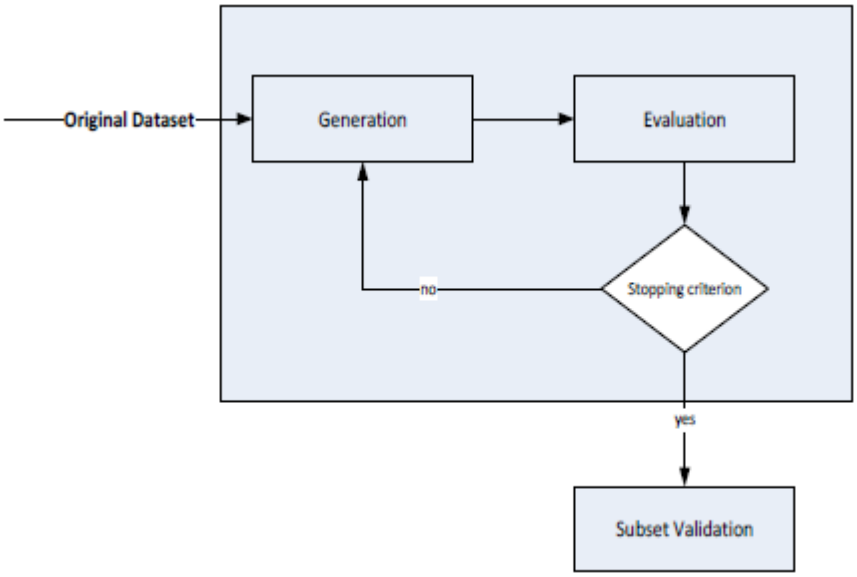
yaitu *feature selection*, untuk mengurangi tingginya dimensi data berupa manipulasi feature sehingga dapat meningkatkan efektifitas dari *classifier*.

Saat ini, ada banyak *measurement function* dalam proses *feature selection* yang dapat digunakan untuk kategorisasi teks. Beberapa diantaranya yaitu *CHI*, *Information Gain*, *Expected Cross Entropy*, dan *Weight of evidence*. Selain itu, ada pula modifikasi dari *Gini Index* agar dapat digunakan langsung sebagai fungsi pada *text feature selection*.

Dari segi implementasi *feature selection*, ada beberapa pendekatan yang dapat digunakan. Salah satunya adalah *filter-based feature selection*. Ini merupakan teknik pemilihan atribut yang tidak bergantung terhadap *classifier* sehingga hasilnya dapat digunakan oleh algoritma *classifier* manapun bahkan oleh algoritma *classifier* yang kompleks seperti *Neural Network*. Selain itu, komputasi dari pendekatan ini relatif rendah sehingga tidak memakan *cost* yang banyak.

Feature Selection atau seleksi fitur adalah sebuah proses yang biasa digunakan pada Machine Learning dimana sekumpulan dari fitur yang dimiliki oleh data digunakan untuk pembelajaran algoritma. Feature selection telah menjadi bidang penelitian aktif dalam pengenalan pola, statistik, dan Data Mining. Seleksi fitur adalah salah satu faktor yang paling penting yang dapat mempengaruhi tingkat akurasi klasifikasi karena jika dataset berisi sejumlah fitur, dimensi dataset akan menjadi besar hal ini membuat rendahnya nilai akurasi klasifikasi. Masalah dalam seleksi fitur adalah pengurangan dimensi, dimana awalanya semua atribut diperlukan untuk memperoleh akurasi yang maksimal.

Ide utama dari Feature Selection adalah memilih subset dari fitur yang ada tanpa transformasi karena tidak semua fitur/atribut relevan dengan masalah. Bahkan beberapa dari fitur atau atribut tersebut mengganggu dan mengurangi akurasi. Noisy Features atau fitur yang tidak terpakai tersebut harus dihapus untuk meningkatkan akurasi. Selain itu dengan fitur atau atribut yang sangatbanyak akan memperlambat proses komputasi. Berikut gambar tahapan Feature Selection.



Gambar 2.1 Feature Selection

Feature Reduction adalah suatu kegiatan yang umumnya bisa dilakukan secara preprocessing dan bertujuan untuk memilih feature yang berpengaruh dan mengesampingkan feature yang tidak berpengaruh dalam suatu kegiatan pemodelan atau penganalisaan data. Ada banyak alternatif yang bisa digunakan dan harus dicoba-coba untuk mencari yang cocok. Secara garis besar ada dua kelompok besar dalam pelaksanaan feature selection: Ranking Selection dan Subset Selection.

Ranking selection secara khusus memberikan ranking pada setiap feature yang ada dan mengesampingkan feature yang tidak memenuhi standar tertentu. Ranking selection menentukan tingkat ranking secara independent antara satu feature dengan feature yang lainnya. Feature yang mempunyai ranking tinggi akan digunakan dan yang rendah akan dikesampingkan. Ranking selection ini biasanya menggunakan beberapa cara dalam memberikan nilai ranking pada setiap feature misalnya regression, correlation, mutual information dan lain-lain.

Subset selection adalah metode selection yang mencari suatu set dari features yang dianggap sebagai optimal feature. Ada tiga jenis metode yang bisa digunakan yaitu selection dengan tipe wrapper, selection dengan tipe filter dan selection dengan tipe embedded.

Feature Selection Tipe Wrapper: feature selection tipe wrapper ini melakukan feature selection dengan melakukan pemilihan bersamaan dengan pelaksanaan pemodelan. Selection tipe ini menggunakan suatu criterion yang memanfaatkan classification rate dari metode pengklasifikasian/pemodelan yang digunakan. Untuk mengurangi computational cost, proses pemilihan umumnya dilakukan dengan

memanfaatkan classification rate dari metode pengklasifikasian/pemodelan untuk pemodelan dengan nilai terendah (misalnya dalam kNN, menggunakan nilai k terendah). Untuk tipe wrapper, perlu untuk terlebih dahulu melakukan feature subset selection sebelum menentukan subset mana yang merupakan subset dengan ranking terbaik. Feature subset selection bisa dilakukan dengan memanfaatkan metode sequential forward selection (dari satu menjadi banyak feature), sequential backward selection (dari banyak menjadi satu), sequential floating selection (bisa dari mana saja), GA, Greedy Search, Hill Climbing, Simulated Annealing, among others.

Feature Selection Tipe Filter: feature selection dengan tipe filter hampir sama dengan selection tipe wrapper dengan menggunakan intrinsic statistical properties dari data. Tipe filter berbeda dari tipe wrapper dalam hal pengkajian feature yang tidak dilakukan bersamaan dengan pemodelan yang dilakukan. Selection ini dilakukan dengan memanfaatkan salah satu dari beberapa jenis filter yang ada. Contohnya: Individual Merit-Base Feature Selection dengan selection criterion: Fisher Criterion, Bhattacharyya, Mahalanobis Distance atau Divergence, Kullback-Leibler Distance, Entropy dan lain-lain. Metode filter ini memilih umumnya dilakukan pada tahapan preprocessing dan mempunyai computational cost yang rendah.

Feature Selection Tipe Embedded: feature selection jenis ini memanfaatkan suatu learning machine dalam proses feature selection. Dalam sistem selection ini, feature secara natural dihilangkan, apabila learning machine menganggap feature tersebut tidak begitu berpengaruh. Beberapa learning machine yang bisa digunakan antara lain: Decision Trees, Random Forests dan lain-lain.

Why Do Feature Selection?

Feature selection is critical to building a good model for several reasons. One is that feature selection implies some degree of *cardinality reduction*, to impose a cutoff on the number of attributes that can be considered when building a model. Data almost always contains more information than is needed to build the model, or the wrong kind of information. For example, you might have a dataset with 500 columns that describe the characteristics of customers; however, if the data in some of the columns is very sparse you would gain very little benefit from adding them to the model, and if some of the columns duplicate each other, using both columns could affect the model.

Not only does feature selection improve the quality of the model, it also makes the process of modeling more efficient. If you use unneeded columns while building a model, more CPU and memory are required during the training process, and more storage space is required for the completed model. Even if resources were not an issue, you would still want to perform feature selection and identify the best columns, because unneeded columns can degrade the quality of the model in several ways:

a. Noisy or redundant data makes it more difficult to discover meaningful patterns.

b. If the data set is high-dimensional, most data mining algorithms require a much larger training data set.

During the process of feature selection, either the analyst or the modeling tool or algorithm actively selects or discards attributes based on their usefulness for analysis. The analyst might perform feature engineering to add features, and remove or modify existing data, while the machine learning algorithm typically scores columns and validates their usefulness in the model.

In short, feature selection helps solve two problems: having too much data that is of little value, or having too little data that is of high value. Your goal in feature selection should be to identify the minimum number of columns from the data source that are significant in building a model.

How Feature Selection Works in SQL Server Data Mining

Feature selection is always performed before the model is trained. With some algorithms, feature selection techniques are "built-in" so that irrelevant columns are excluded and the best features are automatically discovered. Each algorithm has its own set of default techniques for intelligently applying feature reduction. However, you can also manually set parameters to influence feature selection behavior.

During automatic feature selection, a score is calculated for each attribute, and only the attributes that have the best scores are selected for the model. You can also adjust the threshold for the top scores. SQL Server Data Mining provides multiple methods for calculating these scores, and the exact method that is applied in any model depends on these factors:

a. The algorithm used in your model
b. The data type of the attribute
c. Any parameters that you may have set on your model

Feature selection is applied to inputs, predictable attributes, or to states in a column. When scoring for feature selection is complete, only the attributes and states that the algorithm selects are included in the model-building process and can be used for prediction. If you choose a predictable attribute that does not meet the threshold for feature selection the attribute can still be used for prediction, but the predictions will be based solely on the global statistics that exist in the model.

Feature selection affects only the columns that are used in the model, and has no effect on storage of the mining structure. The columns that you leave out of the mining model are still available in the structure, and data in the mining structure columns will be cached.

Feature Selection Scores

SQL Server Data Mining supports these popular and well-established methods for scoring attributes. The specific method used in any particular algorithm or data set depends on the data types, and the column usage.

a. The *interestingness* score is used to rank and sort attributes in columns that contain nonbinary continuous numeric data.
b. *Shannon's entropy* and two *Bayesian* scores are available for columns that contain discrete and discretized data. However, if the model contains any continuous columns, the interestingness score will be used to assess all input columns, to ensure consistency.

Interestingness score

A feature is interesting if it tells you some useful piece of information. However, *interestingness* can be measured in many ways. *Novelty* might be valuable for outlier detection, but the ability to discriminate between closely related items, or *discriminating weight*, might be more interesting for classification. The measure of interestingness that is used in SQL Server Data Mining is *entropy-based*, meaning that attributes with random distributions have higher entropy and lower information gain; therefore, such attributes are less interesting. The entropy for any particular attribute is compared to the entropy of all other attributes, as follows:

Interestingness(Attribute) = - (m - Entropy(Attribute)) * (m - Entropy(Attribute))

Central entropy, or m, means the entropy of the entire feature set. By subtracting the entropy of the target attribute from the central entropy, you can assess how much information the attribute provides. This score is used by default whenever the column contains nonbinary continuous numeric data.

Shannon's Entropy

Shannon's entropy measures the uncertainty of a random variable for a particular outcome. For example, the entropy of a coin toss can be represented as a function of the probability of it coming up heads. Analysis Services uses the following formula to calculate Shannon's entropy:

$$H(X) = -\sum P(xi) \log(P(xi))$$

This scoring method is available for discrete and discretized attributes.

Bayesian with K2 Prior

SQL Server Data Mining provides two feature selection scores that are based on Bayesian networks. A Bayesian network is a *directed* or *acyclic* graph of states and transitions between states, meaning that some states are always prior to the current state, some states are posterior, and the graph does not repeat or loop. By definition, Bayesian networks allow the use of prior knowledge. However, the question of which prior states to use in calculating probabilities of later states is important for algorithm design, performance, and accuracy.

The K2 algorithm for learning from a Bayesian network was developed by Cooper and Herskovits and is often used in data mining. It is scalable and can analyze multiple variables, but requires ordering on variables used as input. For more information, see Learning Bayesian Networks by Chickering, Geiger, and Heckerman.

This scoring method is available for discrete and discretized attributes.

Bayesian Dirichlet Equivalent with Uniform Prior

The Bayesian Dirichlet Equivalent (BDE) score also uses Bayesian analysis to evaluate a network given a dataset. The BDE scoring method was developed by Heckerman and is based on the BD metric developed by Cooper and Herskovits. The Dirichlet distribution is a multinomial distribution that describes the conditional probability of each variable in the network, and has many properties that are useful for learning.

The Bayesian Dirichlet Equivalent with Uniform Prior (BDEU) method assumes a special case of the Dirichlet distribution, in which a mathematical constant is used to create a fixed or uniform distribution of prior states. The BDE score also assumes likelihood equivalence, which means that the data cannot be expected to discriminate equivalent structures. In other words, if the score for If A Then B is the same as the score for If B Then A, the structures cannot be distinguished based on the data, and causation cannot be inferred.

For more information about Bayesian networks and the implementation of these scoring methods, see Learning Bayesian Networks.

Feature Selection Methods per Algorithm

The following table lists the algorithms that support feature selection, the feature selection methods used by the algorithm, and the parameters that you set to control feature selection behavior:

Algorithm	Method of analysis	Comments
Naive Bayes	Shannon's Entropy Bayesian with K2 Prior Bayesian Dirichlet with uniform prior (default)	The Microsoft Naïve Bayes algorithm accepts only discrete or discretized attributes; therefore, it cannot use the interestingness score. For more information about this algorithm, see Microsoft Naive Bayes Algorithm Technical Reference.
Decision trees	Interestingness score Shannon's Entropy Bayesian with K2 Prior Bayesian Dirichlet with uniform prior (default)	If any columns contain non-binary continuous values, the interestingness score is used for all columns, to ensure consistency. Otherwise, the default feature selection method is used, or the method that you specified when you created the model. For more information about this algorithm, see Microsoft Decision Trees Algorithm Technical Reference.
Neural network	Interestingness score Shannon's Entropy Bayesian with K2 Prior Bayesian Dirichlet with uniform prior (default)	The Microsoft Neural Networks algorithm can use both Bayesian and entropy-based methods, as long as the data contains continuous columns. For more information about this algorithm, see Microsoft Neural Network Algorithm Technical Reference.
Logistic regression	Interestingness score Shannon's Entropy Bayesian with K2 Prior Bayesian Dirichlet	Although the Microsoft Logistic Regression algorithm is based on the Microsoft Neural Network algorithm, you cannot customize logistic regression models to control feature selection behavior; therefore, feature selection always default to the method that is most appropriate for the attribute. If all attributes are discrete or discretized, the default is BDEU.

Algorithm	Method of analysis	Comments
	with uniform prior (default)	For more information about this algorithm, see Microsoft Logistic Regression Algorithm Technical Reference.
Clustering	Interestingness score	The Microsoft Clustering algorithm can use discrete or discretized data. However, because the score of each attribute is calculated as a distance and is represented as a continuous number, the interestingness score must be used. For more information about this algorithm, see Microsoft Clustering Algorithm Technical Reference.
Linear regression	Interestingness score	The Microsoft Linear Regression algorithm can only use the interestingness score, because it only supports continuous columns. For more information about this algorithm, see Microsoft Linear Regression Algorithm Technical Reference.
Association rules Sequence clustering	Not used	Feature selection is not invoked with these algorithms. However, you can control the behavior of the algorithm and reduce the size of input data if necessary by setting the value of the parameters MINIMUM_SUPPORT and MINIMUM_PROBABILIITY. For more information, see Microsoft Association Algorithm Technical Reference and Microsoft Sequence Clustering Algorithm Technical Reference.
Time series	Not used	Feature selection does not apply to time series models. For more information about this algorithm, see Microsoft Time Series Algorithm Technical Reference.

Feature selection merupakan bagian penting untuk mengoptimalkan kinerja dari *classifier. Feature selection* dapat didasarkan pada pengurangan ruang fitur yang besar, misalnya dengan mengeliminasi atribut yang kurang relevan. Penggunaan algoritma *feature selection* yang tepat dapat meningkatkan *accuracy*. Algoritma *feature selection* dapat dibedakan menjadi dua tipe, yaitu *filter* dan *wrapper*. Contoh dari tipe *filter* adalah *information gain* (IG), *chi-square*, dan *log likelihood ratio*. Contoh dari tipe *wrapper* adalah *forward selection* dan *backward elimination.* Hasil *precision* dari tipe *wrapper* lebih tinggi daripada tipe *filter*, tetapi hasil ini tercapai dengan tingkat kompleksitas yang besar. Masalah kompleksitas yang tinggi juga dapat menimbulkan masalah.

TEXT MINING

Dalam ilmu Data Mining di kenal dengan istilah Text Mining. Text mining atau text analytics adalah istilah yang mendeskripsikan sebuah teknologi yang mampu menganalisis data teks semi-terstruktur maupun tidak terstruktur, hal inilah yang membedakannya dengan data mining dimana data mining mengolah data yang sifatnya terstruktur. Text mining adalah sebuah penelitian baru yang menarik yang mencoba memecahkan masalah informasi yang overload, dengan menggunakan teknik dari data mining, machine learning, natural language processing (NLP), information retrieval (IR), dan knowledge management. Text mining melibatkan preprocessing koleksi dokumen (kategorisasi teks, ekstraksi informasi, ekstraksi istilah), penyimpanan representasi menengah, teknik untuk menganalisis representasi menengah ini (seperti analisis distribusi, pengelompokan, analisis tren, dan peraturan asosiasi), dan visualisasi hasilnya.

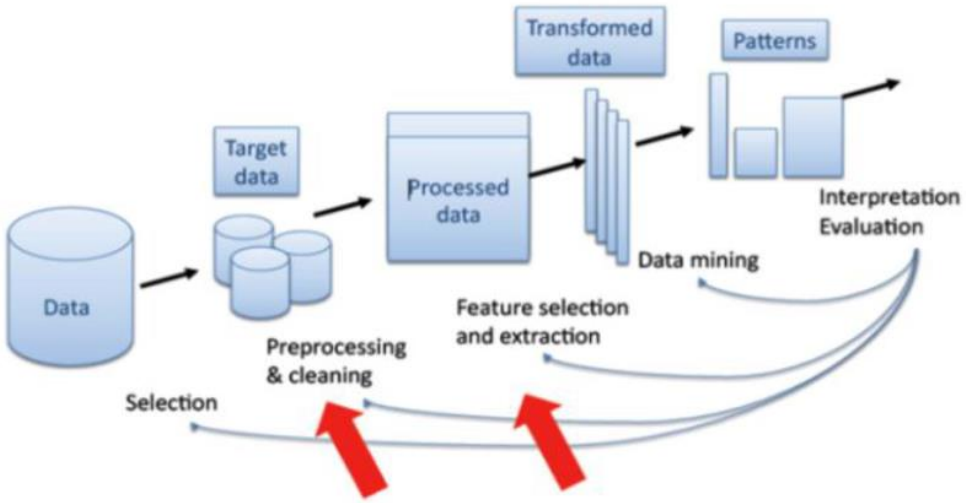
Dalam melakukan text mining, teks dokumen yang digunakan harus dipersiapkan terlebih dahulu, setelah itu baru dapat digunakan untuk proses utama. Proses mempersiapkan teks dokumen atau dataset mentah disebut juga dengan proses *text preprocessing*. *Text preprocessing* berfungsi untuk mengubah data teks yang tidak terstruktur atau sembarang menjadi data yang terstruktur.

Teks yang dilakukan proses text mining pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi yang tinggi, terdapat noise pada data, dan terdapat struktur teks yang tidak baik. Cara yang digunakan dalam mempelajari struktur data teks adalah dengan terlebih dahulu menentukan fitur-fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen, sebelum menentukan fitur-fitur yang mewakili, diperlukan tahap pre-processing. Tujuan utama text preprocessing adalah untuk mendapatkan bentuk data siap olah untuk diproses oleh data mining dari data awal yang berupa data tekstual. Adapun tahap-tahap text preprocessing yang dilakukan adalah sebagai berikut:

1. Case folding, merupakan proses pengubahan huruf dalam dokumen menjadi satu bentuk, misalnya huruf kapital menjadi huruf kecil dan sebaliknya.
2. Tokenizing, merupakan proses pemisahan teks menjadi potongan kalimat dan kata yang disebut token.
3. Filtering, merupakan proses membuang kata-kata serta tanda-tanda yang tidak bermakna secara signifikan, seperti hashtag (#), url, tanda baca tertentu (emoticon), dan lainnya.
4. Stemming, merupakan proses pengubahan kata ke dalam bentuk kata dasar, sehingga berfungsi mengurangi jumlah indeks yang berbeda dari suatu dokumen.

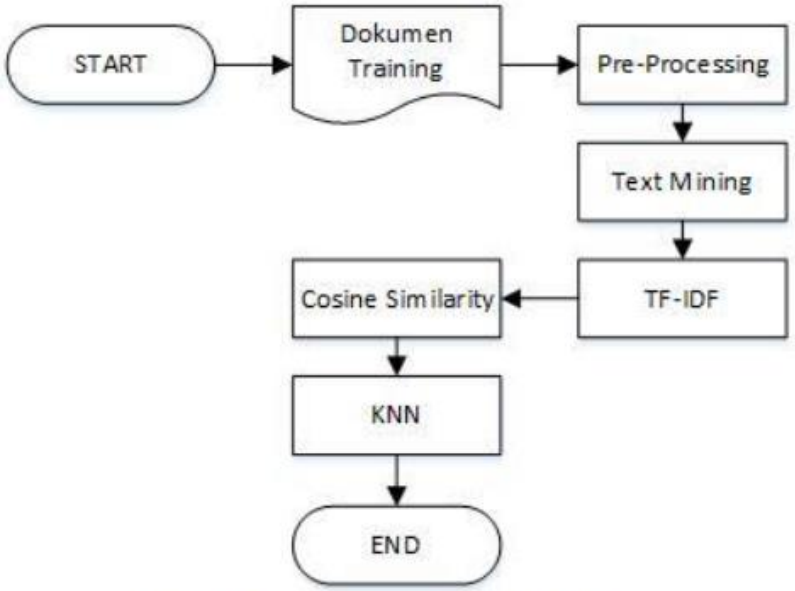
Dalam bidang komputerisasi yang termasuk kedalam *machine learning*, *Naïve Bayes* dan *Support Vector Machine* (SVM) merupakan metode yang digunakan untuk klasifikasi teks dalam *text mining*. Sebagai salah satu metode komputasi yang efisien dan mempunyai *performance preditictive* yang baik, *naïve bayes* merupakan salah satu

metode klasifikasi teks yang popular. *Naïve Bayes* merupakan algoritme yang sering digunakan dalam pengkategorian teks, dimana konsep dasarnya adalah menggabungkan probabilitas kata-kata dan kategori sebuah dokumen.



Gambar 7: Proses *Data Mining*

Pada **Tahap Pertama**: Melakukan persiapan data dalam dari log sistem yang ada dalam aplikasi server pemberitaan. Data dipilih dari sekian banyak data log khusus bagian log transaksi sistem sms.
Tahap Kedua: Melakukan pre-prosesing dengan text mining yang meliputi *tokenizing, filtering, steaming* dan *stopward* sehingga data siap diolah ke proses berikutnya.
Tahap Ketiga: Melakukan pengolahan data dengan menggunakan algoritma data mining yang terdiri dari, estimasi, prediksi, klasifikasi, cluster dan asosiasi.
Tahap Keempat menentukan dictionary manual terkait dengan pemisahan content isi SMS terdiri dari Label Kerja dan tidak Kerja.
Tahap Kelima: Melakukan evaluasi komparasi dari hasil untuk mengukur tingkat akurasi kerja proses text mining dengan KNN terhadap proses manualnya.



Gambar 1. Proses Text *Mining*

Pada tahap pengolahan text *mining* dilakukan tahap *preprocessing* yang dijelaskan pada Gambar 2.

Gambar 2. *Preprocessing* Text *Mining*

Text mining adalah penggalian informasi dari teks oleh user menggunakan *tools* analisis. Secara umum *text mining* mengadopsi proses-proses didalam data *mining* dan didalam *text mining* juga menggunakan teknik data *mining*.

Text preprocessing menjadi tahap awal dalam *text mining*. *Preprocessing* dilakukan untuk menghilangkan bagian atau teks yang tidak diperlukan sehingga mendapatkan data yang berkualitas untuk dieksekusi. Pertama dalam tahap *preprocessing* yaitu *tokenizing* yang bertujuan untuk memecah kalimat menjadi perkata yang terpisah dikenal dengan nama *term* atau token. Selanjutnya *filtering* dengan melakukan penghapusan tanda baca, merubah huruf capital menjadi huruf kecil dan penghapusan *stopword* yang bertujuan untuk menghapus kata-kata yang tidak bermanfaat atau tidak memiliki pengaruh dalam proses. Terakhir yaitu *stemming* untuk mendapatkan kata dasar dari kata yang telah mendapatkan imbuhan atau keterangan lainnya. *Stemming* yang digunakan yaitu *stemming* Nazief dan Adriani karena Algoritma ini memiliki akurasi lebih besar dibandingkan dengan algoritma porter.

Dengan perkembangan teknologi yang semakin besar maka kebutuhan akan penyajian informasi yang cepat dan akurat menjadi salah satu focus utama dalam

penelitaian dan pengembangan guna memenuhi kebutuhan informasi yang semakin cepat dan akurat. Data Mining merupakan kompleks teknologi yang berakar pada berbagai disiplin ilmu: matematika, statistik, ilmu komputer, fisika, teknik, biologi, dll, dan dengan beragam aplikasi dalam berbagai macam domain yang berbeda: bisnis, kesehatan, sains dan teknik , dll Pada dasarnya, data mining dapat dilihat sebagai ilmu menjelajahi dataset besar untuk mengekstraksi informasi tersirat, yang sebelumnya tidak diketahui dan berpotensi berguna.

Sedangkan *Text mining* adalah salah satu penambangan informasi yang berguna dari data – data yang berupa tulisan, dokumen atau text dalam bentuk klasifikasi maupun clustering. Text mining masih merupakan bagian dari data mining dimana akan memproses data – data atau text – text serta dokumen – dokumen yang bisa jadi dalam jumlah sangat besar. Untuk memproses data yang sangat besar tentulah akan memakan sumber daya yang tidak sedikit kaitanya dengan pengolahan data tersebut. Disinilah diperukanya sebuah pemrosesan awal atau preprocessing data text tersebut sebelum data tersebut di lakukan proses text mining sesuai algoritma yang akan diterapkan.

Dengan *text mining* maka kita akan melakukan proses mencari atau penggalian informasi yang berguna dari data tekstual. Ini juga merupakan salah satu kajian penelitian yang sangat menarik dan juga sangat berguna di kemudian hari dimana seperti mencoba untuk menemukan pengetahuan dari dokumen–dokumen atau teks - teks yang tidak terstruktur. *Text mining* sekarang juga memiliki peran yang semakin penting dalam negara berkembangaplikasi, seperti mengetahui isi dari teks secara langsung dari proses *text mining* tanpa perlu membaca satu persatu teks atau tulisan yang ada. Proses Text mining adalah sama dengan data mining, kecuali, beberapa metode dan data yang di kelola nya seperti data teks yang tidak terstruktur, terstruktur sebagian maupun terstruktur seperti teks email, teks HTML, maupun teks komentar serta dari berbagai sumber.

Untuk dapat melakukan penambangan informasi atau text mining maka perlu dilakukan beberapa tahapan yang harus dilakukan untuk mengolah sumber data baik yang terstruktur, terstuktur sebagian dan yang tidak terstruktur dari beberapa sumber maka data-data tersebut perlu dilakukan proses awal atau di sebut sebagai preprocessing text yang bermaksud mengolah data awal yang masih bermacam – macam untuk dijadikan sebuah data teratur yang dapat dikenai atau diterapkan beberapa metode text mining yang ada.

Apa sih arti text mining yang sebenarnya? Definisi akan text mining sudah sering di berikan oleh banyak ahli riset dan praktisi. Seperti hal nya data mining, text mining adalah proses penemuan akan informasi atau trend baru yang sebelumnya tidak terungkap dengan memproses dan menganalisa data dalam jumlah besar. Dalam menganalisa sebagian atau keseluruhan unstructured text, text mining mencoba untuk mengasosiasikan satu bagian text dengan yang lainnya berdasarkan aturanaturan tertentu. Hasil yang di harapkan adalah informasi baru atau “insight” yang tidak terungkap jelas sebelumnya. Seperti halnya data mining, text mining juga menghadapi

masalah yang sama, termasuk jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, dan data “noise.” Berbeda dengan data mining yang utamanya memproses structured data, data yang digunakan text mining pada umumnya dalam bentuk unstructured, atau minimal semistructured, text. Akibatnya, text mining mempunyai tantangan tambahan yang tidak di temui di data mining, seperti struktur text yang complex dan tidak lengkap, arti yang tidak jelas dan tidak standard, dan bahasa yang berbeda ditambah translasi yang tidak akurat.

Dikarenakan structured data ditujukan agar mudah di proses komputer secara automatic, pre-process data di data mining jauh lebih mudah dilakukan dari pada pada unstructured text. Text di ciptakan bukan untuk di gunakan oleh mesin, tapi untuk dikonsumsi manusia langsung. Karena itu, pada umumnya “Natural Language Processor” digunakan untuk memproses unstructured text. Hearst mempertanyakan pengunaan kata ‘mining’ di data mining dan text mining. Kata ‘mining’ memberikan arti dimana fakta-fakta atau relasirelasi baru dihasilkan dari proses me-‘mining’ data. Dia mengklaim bahwa aktivitas data mining lebih memfocuskan pada penemuan trend dan pattern yang sebenarnya sudah ada. Sedangkan ahli text mining yang lain beranggapan bahwa text mining adalah proses penemuan kembali relasi dan fakta yang terkubur didalam text, dan tidak harus baru.

Ulasan di berikutnya sedikit mengikuti definisi text mining oleh Hearst. Seperti di sebutkan sebelumnya, Text mining telah mengadopsi teknik yang di gunakan di bidang natural language processing dan computational linguistics. Walaupun teknik di computational linguistics bisa dibilang maju dan cukup akurat untuk mengekstrak informasi, tujuan text mining bukan hanya mengekstrak informasi. Melainkan untuk menemukan pattern dan informasi baru yang belum terungkapkan, yang sulit ditemukan tanpa analisa yang dalam. Walau kemampuan komputer untuk mencapai kemampuan untuk memproses text seperti manusia sangat sulit, bila tidak mustahil, telah banyak teknik-teknik baru di computational linguistics yang bisa membantu text mining untuk mencerna text lebih jauh lagi.

Sering kali penguna search engine di Internet menganggap search engine sebagai salah satu implementasi text mining. Andil utama search engine hanyalah menyingkirkan text yang tidak memiliki kata-kunci yang di cari pengunna. Dan lagi penguna search engine mengetahui sebelumnya text seperti apa yang hendak dia cari. Bisa dibilang kalau pencarian seperti ini termasuk dalam “Information Retrieval.” Focus information retreieval adalah menemukan dokumen atau text yang memenuhi kriteria pencari. Text mining lebih memfokuskan pada relasi dan co-existence dari satu dokumen dengan yang lainnya. Walaupun text mining lebih dari information retrieval, text mining telah mengadopsi information retrieval untuk menyaring dan mengurangi jumlah informasi untuk diproses selanjutnya. Methode statistik juga sudah mulai sering di gunakan dan di adopsi di computational linguistics dan information retrieval yang nanti nya bisa memberikan tool yang lebih baik dan akurat untuk text mining.

Banyak juga ahli riset yang menkategorikan document categorization sebagai text mining. Walau kategorisasi dokumen dapat memberikan label dan kesimpulan yang

akurat pada dokumen-dokumen tertentu, ini tidak menghasilkan fakta-fakta atau relasi yang baru. Tetapi bilamana label-label atau kesimpulankesimpulan yang di hasilkan di analisa dan di korelasikan lebih lanjut, ini bisa menghasilkan fakta dan relasi baru antara group-group dokumen yang berbeda. Kegiatan seperti ini bisa di masukan dalam text mining.

Contoh dalam mengolah data teks ulasan sebuah hotel dalam bahasa Indonesia yang masih memiliki bentuk tidak terstruktur ke dalam bentuk yang lebih terstruktur dan melakukan ekstraksi informasi dari sejumlah ulasan dengan membuat sebuah *wordcloud. Wordcloud* merupakan kumpulan kata-kata yang paling sering dibicarakan dalam ulasan.

Menggunakan software R Programming untuk melakukan analisis teks, sebelum masuk pada tahap pembuatan *wordcloud,* terlebih dahulu saya akan melakukan *text preprocessing* untuk cleaning data agar data siap di olah. Untuk melakukan analisis teks, dalam R digunakan beberapa *packages* diantaranya *packages "tm", "RColorBrewer", "wordcloud"* dan *"stringr" .* Untuk menginstall packages tersebut ke dalam program R, dapat dilakukan dengan cara menjalankan script berikut:

```
install.packages("tm")
install.packages("RColorBrewer")
install.packages("wordcloud")
install.packages("stringr")
```

Aktifkan packages dengan perintah "library"

```
library("tm")
library("RColorBrewer")
library("wordcloud")

library("stringr")
```

Kemudian, aktifkan folder kerja yang merupakan tempat penyimpanan file postingan (dalam bentuk .csv) dengan perintah "setwd" dan baca file ke dalam R menggunakan perintah "readLines" seperti berikut:

```
setwd("E://KULIAH")
docs<-readLines("dataulasan.csv")
docs
```

Sehingga akan tampil isi ulasan seperti berikut:

```
> docs
 [1] "\"Kamar hotel cukup luas, bersih, dan nyaman.. memiliki view dan teras mengh$
 [2] "kesini sekitar tahun 2013 hotel cukup bagus kamar luas dan nyamansarapan len$
 [3] "\"Hotelnya cukup tenang, nyaman untuk beristirahat. Pilihan menu sarapannya $
 [4] "\"Menyenangkan, lokasi yang strategis, dekat mall, kuliner, kolam renang yan$
 [5] "terakhir memasuki royal ambarukmo adalah saat masih bernama ambarukmo.. tahu$
 [6] "\"Business and Leisure combines together.Staff hotel nya ramah-ramah dan tan$
 [7] "\"Sarapan pagi lengkap and enak, kolam renang and gymnya bersih dan ok, suas$
 [8] "\"Saya puas dengan segala fasilitas yang ada disini, sangat memuaskan. Pool $
 [9] "ketika anda di yogya dan bingung mau nginep dimana...Hotel ini menyediakan a$
[10] "Reception kurang profesional tidak mengerti membaca Remark apa yang di jual $
[11] "\"Hotelnya bagus, bersih, sangat nyaman buat keluarga menginap. Staffnya jug$
[12] "\"Satu kata untuk hotel ini yaitu terbaik. Hotel yang asri, banyak tanaman a$
[13] "\"Royal Ambarukmo mempunyai sejarah yang menarik,bahkan sebagian dari bangun$
[14] "Hotel yang memberikan pelayanan breakfast yang enak dengan variasi yang bera$
```

Data ulasan di atas masih berbentuk tidak terstruktur, dan masih banyak noise sehingga perlu dilakukan cleaning data. Untuk melakukan cleaning data terlebih dahulu data teks harus di ubah ke dalam bentuk Corpus dengan menjalankan script berikut:

```
docs <- Corpus(VectorSource(docs))
```

Kemudian, akan dilakukan pembersihan data, dengan mengganti tanda “/”, “@” and “|” dengan sebuah spasi menggunakan perintah:

```
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/")
docs <- tm_map(docs, toSpace, "@")
docs <- tm_map(docs, toSpace, "\\|")
```

Kemudian dilakukan proses *case folding,* yakni menyeragamkan huruf ke dalam bentuk huruf kecil menggunakan perintah

```
docs <- tm_map(docs, content_transformer(tolower))
```

Kemudian menghapus tanda baca (punctuation) dengan menggunakan perintah:

```
docs <- tm_map(docs, toSpace, "[[:punct:]]")
```

Menghapus angka dengan menggunakan perintah:

```
docs <- tm_map(docs, toSpace, "[[:digit:]]")
```

Kemudian dilakukan proses filtering yakni membuang daftar kata-kata yang kurang penting untuk di analisis menggunakan stopwords. Stopword / stoplist adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Untuk

menjalankan stopwords dapat dilakukan dengan menjalankan perintah berikut:

```
myStopwords = readLines("stopword_id.csv")
docs <- tm_map(docs, removeWords, myStopwords)
```

Untuk menghapus daftar kata secara manual juga dapat dilakukan dengan cara berikut:

```
docs <- tm_map(docs, removeWords, c("you","also","hotel","ambarrukmo","royal"))
```

Menghapus spasi yang tidak berguna, yakni terdapat spasi yang berlebih pada selah antara dua kata, untuk menghapus spasi berlebih tersebut dapat digunakan perintah berikut:

```
docs <- tm_map(docs, stripWhitespace)
```

Menghapus URL web dengan menjalankan perintah:

```
removeURL <- function(x) gsub("http[[:alnum:]]*", " ", x)
docs <- tm_map(docs, removeURL)
```

Untuk memperbaiki kata-kata yang salah (spelling normalization), dapat dilakukan dengan cara manual menggunakan perintah:

```
docs <- tm_map(docs, gsub, pattern="Howver", replacement="However")
docs <- tm_map(docs, gsub, pattern="goood", replacement="good")
```

Setelah melalui tahap cleaning data, kemudian merubah data ke dalam bentuk Term Document Matrix, dan mengubah ke dalam bentuk data frame sehingga dapat dihitung frekuensi setiap kata. Adapun perintah yang digunakan adalah sebagai berikut:

```
dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 10)
```

Sehingga akan tampil jumlah frekuensi kata seperti berikut:

```
> dtm <- TermDocumentMatrix(docs)
> m <- as.matrix(dtm)
> v <- sort(rowSums(m),decreasing=TRUE)
> d <- data.frame(word = names(v),freq=v)
> head(d, 10)
        word freq
kolam   kolam   12
kamar   kamar    8
luas     luas    8
renang renang    8
nyaman nyaman    7
mall     mall    6
enak     enak    6
bagus   bagus    5
ramah   ramah    5
and       and    5
```

Setelah di peroleh frekuensi setiap kata, kemudian kata-kata tersebut dapat di ubah ke dalam bentuk wordcloud menggunakan perintah berikut:

```
dtm <- TermDocumentMatrix(docs)
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
          max.words=50, random.order=FALSE, rot.per=0.35,

    colors=brewer.pal(8, "Dark2"))
```

Sehingga akan muncul tampilan wordcloud seperti gambar berikut:

Berdasarkan gambar wordcloud diatas, semakin besar kata pada tampilan wordcloud maka menunjukkan semakin besar pula frekuensi kata tersebut dalam dokumen ulasan.

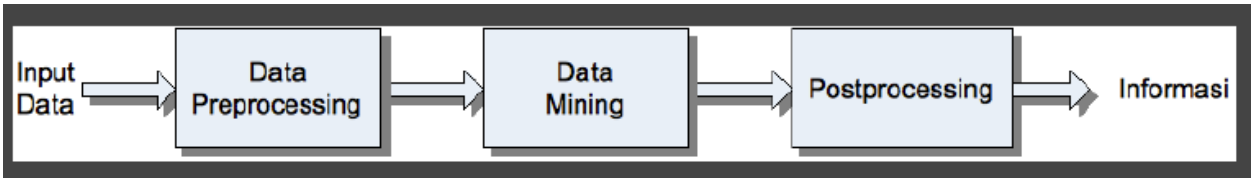
Aplikasi text mining

Aplikasi text mining bisa di bagi berdasarkan tipe unstructured text yang di proses. Untuk ustructured text dalam bentuk emails, intant messages, dan blogs, pada umumnya pengguna ingin mencari atau “mine” informasi mengenai orang (seperti email pengirim, alamat, nama lengkap, dll), perusahaan (seperti nama lengkap dan lokasi), organisasi, dan kejadian-kejadian (seperti penemuan baru, pengumuman penting, dll). Untuk berita dari berbagai sumber, text mining bisa di gunakan untuk membandingkan berita yang sama atau berbeda yang berasal dari sumber yang berbeda, mungkin dengan bahasa yang berbeda. Lebih jauh lagi adalah analisa dan organisasi isi berita berdasarkan waktu publikasi (atau “temporal analysis”). Text mining juga bisa membantu untuk proses “deduplication” di sini. Untuk buku-buku dan artikel-artikel science, text mining di butuhkan untuk mendeteksi trend di bidang riset tertentu. Salah satu cara yang bisa di lakukan adalah dengan memonitor jumlah publikasi untuk bidang riset tertentu untuk jangka waktu tertentu. Hasil-hasil untuk bidang riset yang berbeda bisa di bandingkan dan di analisa guna memberikan hasil trend yang berarti. Untuk technical working paper, documentasi, dan software specifikasi dokumen, text mining bisa di gunakan untuk mengekstrak software requirement dari spesifikasi dokumen secara otomatis atau mendeteksi ke kurangan antara source code dan documentasinya secara otomatis. For web pages, text mining bisa di gunakan untuk menganalisa website perusahaan, structur websitenya, perbandingan website content yang satu dengan site yang lain. Masih banyak lagi aplikasi text mining yang di butuhkan.

Proses Text Mining

Proses text mining mencakup beberapa sub-task, seperti information retrieval, categorization, POS tagging, Clustering, dan lainnya, yang bisa di katergorikan kedalam framework “Knowledge Discovery in Databases” (KDD), yang tidak lain adalah proses mengidentifikasikan pattern di dalam data yang benar, unik, berguna, dan dimengerti. KDD proses interaktif, bisa berulang, dan terdiri dari step Selection, Preprocessing, Transformation, Data Mining, dan Interpretation/Evaluation. Dalam sesi ini, proses dan kegiatan text mining yang beragam akan saya coba assosiasikan dengan KDD step dan ulas secara singkat.

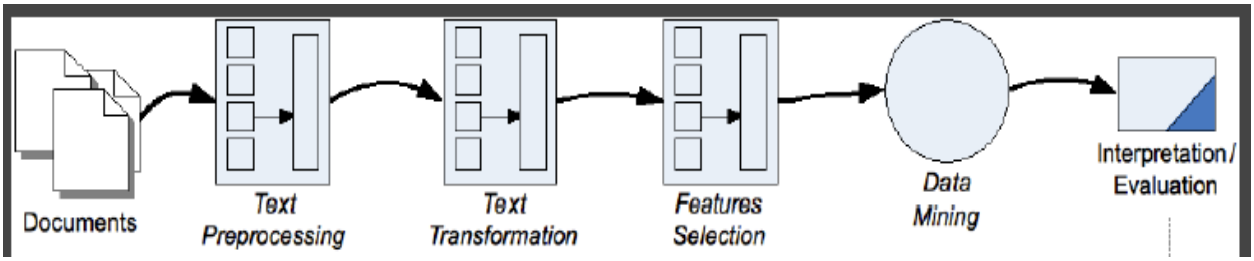
Data mining adalah suatu proses yang secara otomatis mencari atau **menemukan informasi** yang bermanfaat dan suatu kumpulan data yang besar. Data Mining lebih dekat pada bidang **pencarian pengetahuan** dalam basis data (knowledge discovery in database / KDD), yang merupakan proses **konversi** dari data mentah menjadi informasi yang bermanfaat.



Data mining dibagi dalam dua kelompok jenis tugas analisis data:

a. Predictive task : bertugas untuk **memprediksi nilai** sebuah atribut tertentu (target) didasarkan pada nilai atribut lain (*explanatory)*
b. Descriptive task : bertugas **mendapatkan pola** analisis asosiasi (*association analysis), pengelompokan (clustering),* penyimpangan (*anomaly detection) yang meringkas* hubungan-hubungan dalam data

Text mining merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks, yaitu proses penganalisisan teks guna menyarikan informasi yang bermanfaat untuk tujuan tertentu. Berdasarkan ketidakteraturan struktur data teks, maka proses text mining memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur.



Tahapan Text Mining

Text mining merupakan penerapan konsep dan teknik data mining untuk **mencari pola dalam teks**. Teks Mining : Proses penganalisisan teks guna **menyarikan informasi** yang bermanfaat untuk tujuan tertentu.

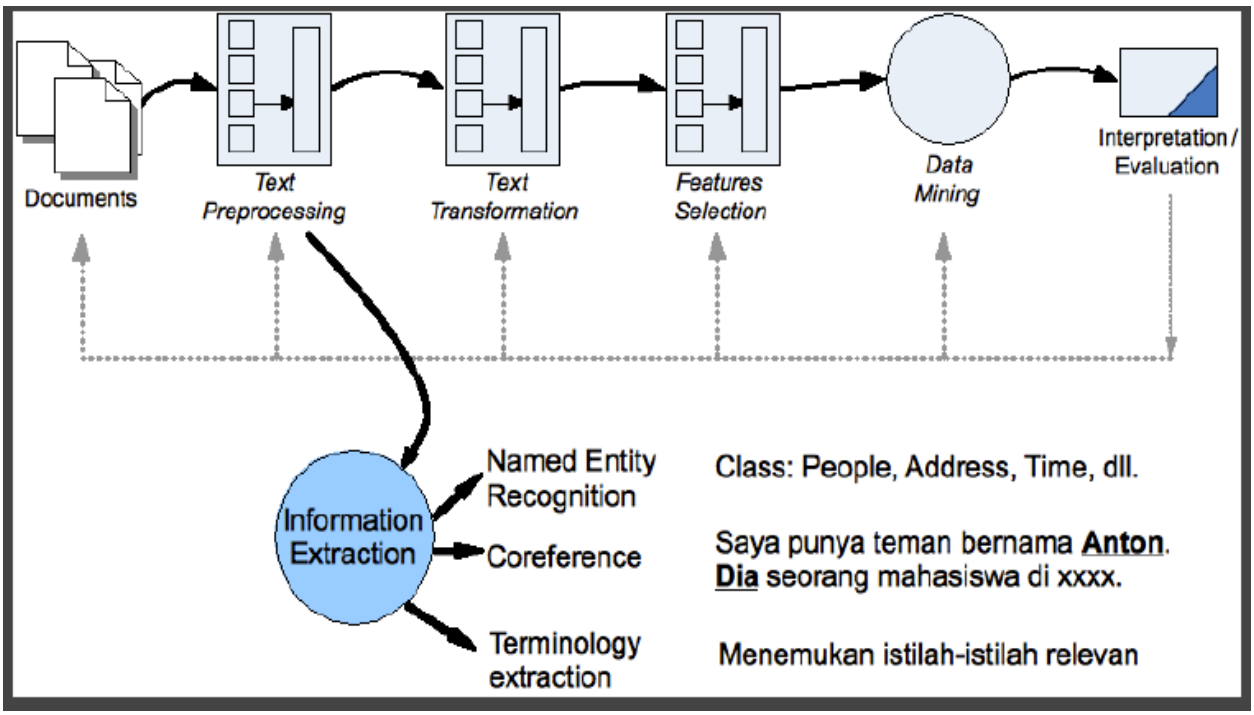
Perbedaan mendasar dengan Data Mining pada umumnya, **Text Mining** mengolah data **teks yang tidak terstruktur**, maka proses text mining memerlukan beberapa tahap awal ***(preprocessing)*** yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur.

Perbedaan	Data Mining	Text Mining
Data Object	Numerical & categorical data	Textual data
Data structure	Structured	Unstructured & semi-structured
Data representation	Straightforward	Complex
Space dimension	< tens of thousands	> tens of thousands
Methods	Data analysis, machine learning, Neural Network, etc.	Data mining, information Retrieval, NLP, etc.
Maturity	Broad implementation since 1994	Broad implementation starting 2000

Masalah umum yang ditangani:

a. Pengorganisasian dan Clustering Dokumen
   Clustering adalah pengorganisasian kumpulan pola ke dalam cluster (kelompok-kelompok) berdasar atas kesamaannya. Pola-pola dalam suatu cluster akan memiliki kesamaan ciri/sifat daripada pola-pola dalam cluster yang lainnya. Clustering bermanfaat untuk melakukan analisis pola-pola yang ada, mengelompokkan, dan membuat keputusan. Metodologi clustering lebih cocok digunakan untuk eksplorasi hubungan antar data untuk membuat suatu penilaian terhadap strukturnya.

b. Klasifikasi Dokumen
   Klasifikasi adalah mengelompokkan dokumen berdasarkan data training yang sudah dilabeli. Perbedaanya dengan clustering adalah pada klasifikasi, kelas/kategorinya sudah ditentukan di awal, sedangkan pada clustering tidak.



c. Information Extraction
   Information Extraction bermanfaat untuk menggali struktur informasi dari sekumpulan dokumen. Dalam menerapkan IE, perlu sekali dilakukan pembatasan domain problem. IE sangat memerlukan NLP untuk mengetahui gramatikal dari setiap kalimat yang ada. Sebagai contoh:

a. “Indonesia dan Singapore menandatangani MoU kerjasama dalam bidang informasi dan komunikasi.”
b. KerjaSama(Indonesia, Singapore, TIK)

Dengan IE, kita dapat menemukan:

a. concepts (CLASS)
b. concept inheritance (SUBCLASS-OF)
c. concept instantiation (INSTANCE-OF)
d. properties/relations (RELATION)
e. domain and range restrictions (DOMAIN/RANGE)
f. equivalence

Web Mining

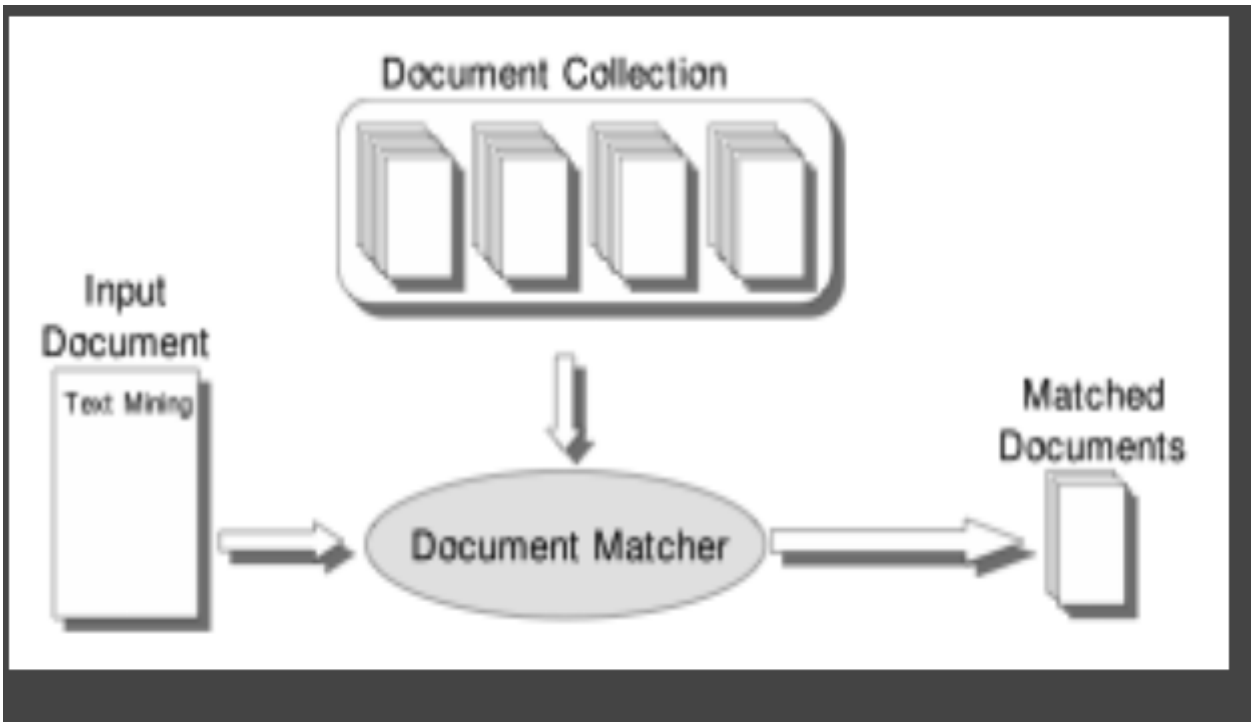
Jumlah data/informasi di web sangat besar dan terus bertambah.

a. tipe data beragam
b. informasi pada web sangat beragam.
c. informasi-informasi di web saling terhubung.
d. informasi di web sangat "kotor".
e. web juga merupakan service.
f. web dinamis
g. web merupakan sarana komunitas sosial virtual.

Web Mining bertujuan untuk menemukan informasi atau pengetahuan dari Web hyperlink structure, contoh :menemukan halaman web terpenting; menemukan komunitas pemakai yang berbagi ketertarikan topik yang sama.

Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah melakukan pengolahan untuk **memahami Bahasa alami** yang diucapkan manusia Bahasa alami adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari user bisa dipahami dengan baik oleh komputer.



Information Retrieval (IR)

Konsep dasar dari IR adalah pengukuran kesamaan sebuah perbandingan antara dua dokumen, mengukur sebearapa mirip keduanya. Setiap input query yang diberikan, dapat dianggap sebagai sebuah dokumen yang akan dicocokan dengan dokumendokumen lain. Pengukuran kemiripan serupa dengan metode klasifikasi yang disebut metode nearest-neighbour.

Perbedaan mendasar antara Text Mining dan IR :

a. Text Mining : Discovery of novel information
   Extracting Ore from otherwise worthless rock : menemukan informasi yang relevan dan bermanfaat dari sekumpulan data besar yang kelihatanya tidak berguna.
b. IR : Retrieval of Non-novel Information
   Finding needles in a needle-stack : mencari informasi yang relevan di antara informasi-informasi lain yang berguna namun tidak relevan

Search Engine merupakan aplikasi nyata dari **Information Retrieval** pada bidang web.

DAFTAR PUSTAKA

https://www.datascience.or.id/detail_artikel/52/supervised-and-unsupervised-learning

Bustami. (2013). Penerapan Algoritma Naïve Bayes Untuk Mengklasifikasi Data Nasabah Asuransi*. TECHSI : Jurnal Penelitian Teknik Informatika*, 3(2), 129-132

Hamzah, A. (2012). Klasifikasi teks dengan naïve bayes classifier (nbc) untuk pengelompokan teks berita dan abstract akademis. In *Prosiding Seminar Nasional*.

Han, J., & Kamber, M. (2006). Data Mining Concept and Tehniques. San Fransisco: Morgan Kauffman.

J.Kittler, “Feature Selection & Extraction”, in Handbook of Pattern Recognition and Image Processing, Tzay Y. Young, King Sun Fu Ed. Academic Press, 1986.

Koncz, P., & Paralic, J. (2011). An approach to feature selection for sentiment analysis. In *2011 15th IEEE International Conference on Intelligent Engineering Systems* (pp. 357–362). IEEE. doi:10.1109/INES.2011.5954773

Prasetyo, Eko. 2012. *DATA MINING – Konsep dan Aplikasi menggunakan MATLAB.* Yogyakarta: Andi.

Sanjaya dan Absar. *Pengelompokkan Dokumen Menggunakan Winnowing Fingerprint dengan Metode K-Nearest Neighbour*. Jurnal CoreIT. 2015; Vol. 1, No. 2, Desember.

Wang, S., Li, D., Song, X., Wei, Y., & Li, H. (2011). A feature selection method based on improved fisher’s discriminant ratio for text sentiment classification. *Expert Systems with Applications*, *38*(7), 8696–8702. doi:10.1016/j.eswa.2011.01.077