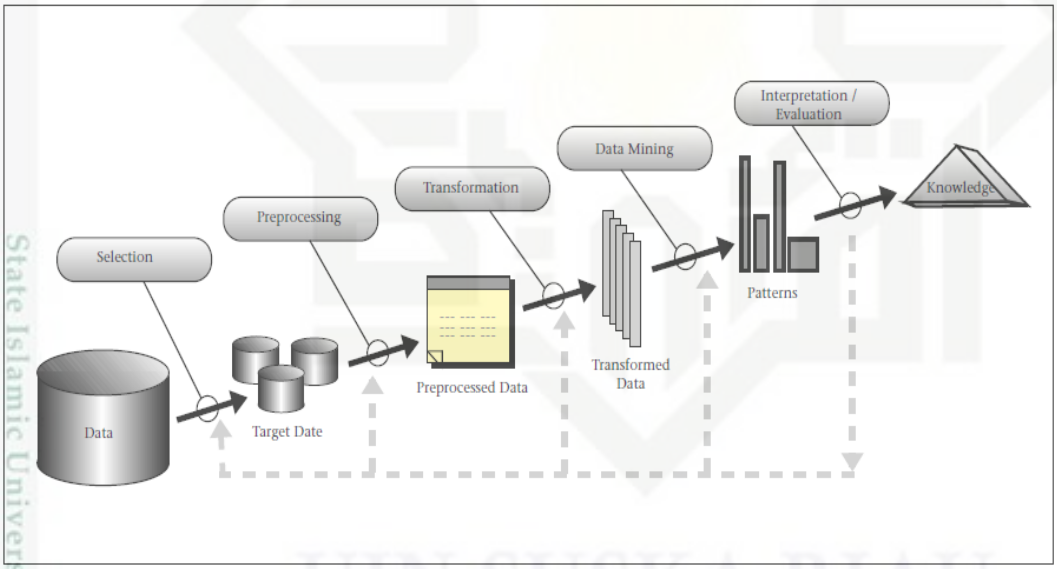
BAB II
LANDASAN TEORI

2.1 *Knowledge Discovering in Data* (KDD)

Istilah *data mining* dan *knowledge discovery in databases* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah data mining. Proses KDD secara garis besar dapat dijelaskan sebagai berikut, Shapiro (sebagaimana dikutip oleh Maihendra, 2016).



Gambar 2.1 Tahapan-tahapan KDD (Sumber : Shapiro (dalam Maihendra, 2016))

2.1.1 *Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional (Maihendra, 2016).

2.1.2 *Preprocessing*

Sebelum data diolah ke tahap selanjutnya, data perlu dilakukan prerocessing terlebih dahulu. Tujuan *preprocessing* adalah agar meningkatkan performance dari teknik atau metode *data mining*. Ada beberapa tahapan *preprocessing* sebagai berikut :

1. Data *Cleaning*

Proses menghilangkan noise dari data yang tidak konsisten atau tidak relevan. Pembersihan data ini akan mempengaruhi performansi teknik/metode data mining karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

2. Data *Integration*

Penggabungan data dari berbagai *database* ke dalam satu *database* baru. Integrasi data dilakukan pada atribut-atribut yang mengidentifikasikan entitas-entitas yang unik. Ilustrasi dalam database seperti *primary key* dan *foreign key*.

2.1.3 *Transformation*

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*. Beberapa metode *data mining* membutuhkan format data yang khusus sebelum bisa diaplikasikan. Cara lain yang dapat dilakukan dalam transformasi data adalah *normalization*, dimana data atribut dibuat dalam skala tertentu sehingga menjadi kisaran data yang lebih kecil sehingga sebaran datanya tidak terlalu jauh. Dengan rumus normalisasi :

$$v^i = \frac{v - min_a}{max_a - min_a} (new\_max_a - new\_min_a) + new\_min_a \qquad (2.1)$$

Dimana :

v^i : Data baru setelah normalisasi
v : Data sebelum normalisasi
new_max_a : Batas nilai max baru adalah 1
new_min_a : Batas nilai min baru adalah 0
max_a : Nilai maximum pada kolom
min_a : Nilai minimum pada kolom

2.1.4 Data Mining

Sebagaimana dikutip oleh Maihendra, (2016) *Data mining* adalah teknik bagaimana menelusuri data yang ada untuk membangun sebuah model, kemudian menggunakan model tersebut agar dapat mengenali pola data lain yang tidak berada dalam basis data yang disimpan. Teknik Data Mining didukung oleh tiga teknologi yaitu pengumpulan data secara besar, multiprocessor pada komputer dan algoritma *data mining* (Kusrini, 2009).

2.1.4.1 Proses Data Mining

Menurut (Gorunescu, 2011) proses *data mining* terbagi dalam tiga aktifitas yaitu (sebagaimana dikutip oleh Maihendra, 2016) :

1. Eksplorasi data, terdiri dari aktifitas pembersihan data dan transformasi data.
2. Membuat model dan pengujian validitas model, merupakan pemilihan terhadap model-model yang sudah dikembangkan yang cocok dengan kasus yang dihadapi. Dengan kata lain, dilakukan pemilihan model secara kompetitif.
3. Penerapan model dengan data baru untuk menghasilkan perkiraan dari kasus yang ada. Tahap ini merupakan tahap yang menentukan apakah model yang dibangun dapat menjawab permasalahan yang dihadapi.

2.1.4.2 Pengelompokan Data Mining

Menurut (Larose, 2005), dalam bukunya yang berjudul ”*Discovering Knowledge in Data:An Introduction to Data Mining*”, *data mining* dibagi menjadi beberapa kelompok berdasarkan tugas atau pekerjaan yang dapat dilakukan (sebagaimana dikutip oleh Maihendra, 2016), yaitu :

a. Deskripsi

Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari pola kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

b. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan baris data (*record*) lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi.Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.

c. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

d. Klasifikasi

Data input untuk klasifikasi adalah koleksi dari *record*. Setiap *record* dikenal sebagai *instance* atau atribut, yang ditentukan oleh sebuah *tuple* (x,y), dimana x adalah himpunan atribut dan y adalah label kelas. Beberapa teknik klasifikasi yang sering digunakan adalah *decision tree classifier, rule-based classifier, neural-network, support vector machine, naive Bayes classifier, K-Nearest Neighbor* dan *Modified K-Nearest Neighbor.*

e. Pengklasteran (*Clustering*)

Pengklasteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Klaster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan *record* dalam klaster yang lain. Berbeda dengan klasifikasi, padapengklasteran tidak ada variabel target. Pengklasteran tidak melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target, akan tetapi, algoritma pengklasteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memilikikemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

f. Asosiasi

Tugas asosiasi dalam *data mining* adalah untuk menemukan atribut yang muncul dalam satu waktu. Salah satu implementasi dari asosiasi adalah *market basket analysis* atau analisis keranjang belanja.

Pada penelitian ini diterapkan teknik klasifikasi yaitu metode *Modified k-Nearest Neighbor (MK-NN)* untuk menemukan pengetahuan baru dari data yang akan diolah.

2.1.4.3 *Feature Selection* (Pemilihan Fitur)

Pemilihan fitur adalah suatu proses yang dilakukan untuk menentukan fitur-fitur yang signifikan dalam *dataset* yang sesuai untuk permasalahan yang akan dipecahkan. Semakin baik hasil pemilihan fitur dapat meningkatkan nilai *accuracy* dari metode deteksi yang diuji. Pemilihan fitur juga bermanfaat dalam mereduksi dimensi dari *dataset* dengan cara 'membuang' fitur-fitur yang tidak signifikan (tidak memiliki pengaruh terhadap penentuan kelas / label). Tujuan utama dari seleksi fitur adalah memperoleh kumpulan fitur-fitur terbaik yang dapat meningkatkan performansi dari model deteksi yang dikembangkan. Beberapa keuntungan dari seleksi fitur adalah :

a. Meminimalkan *overfit* : proses seleksi fitur dapat menghilangkan data *redundan* dan *noise* yang dapat mengakibatkan *overfit* pada proses *clustering*.
b. Meningkatkan *accuracy* : berkaitan dengan (a), proses seleksi fitur akan menghilangkan fitur-fitur yang tidak signifikan yang dapat mengakibatkan *misleading* akibat *overfit*, dengan demikian nilai *accuracy* akan meningkat.
c. Mengurangi waktu pemrosesan : semakin sederhana dimensi dari *dataset* maka algoritma *learning* dapat dijalankan dengan lebih cepat dan efisien.

Proses seleksi fitur antara lain melibatkan kombinasi dari proses pencarian, estimasi pengaruh fitur dalam penentuan label data, dan evaluasi dengan menggunakan algoritma *machine learning*. Dengan demikian seleksi fitur akan melibatkan banyak sekali kemungkinan proses (Mutaqien, 2016). Untuk

mengoptimalkan proses seleksi fitur, digunakan prosedur pencarian secara heuristik yang dipadukan dengan *evaluator* yang berfungsi untuk mengestimasi tingkat pengaruh fitur. Secara umum seleksi fitur dikelompokkan menjadi tiga teknik Mutaqien, (2016), yaitu : teknik *filter*, teknik *wrapper* dan teknik *embedded*.

Teknik *filter* menggunakan pengujian statistik untuk melakukan evaluasi terhadap fitur sehingga teknik ini tidak bergantung kepada algoritma *learning* tertentu. Teknik *filter* akan menghasilkan rangking fitur mulai dari fitur yang paling signifikan sampai yang tidak signifikan. Suatu fitur disebut tidak signifikan jika fitur tersebut tidak memiliki pengaruh dalam penentuan label dari data pada *dataset*. 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.

Teknik *wrapper* melakukan seleksi fitur dengan membentuk *subset-subset* yang terdiri dari kombinasi yang mungkin dari fitur *dataset*. Kemudian masing-masing *subset* tersebut akan dievaluasi dengan algoritma learning untuk mendapatkan tingkat deteksi dalam penentuan label / kelas. Teknik *wrapper* dapat memberikan hasil yang lebih baik daripada teknik filter, namun membutuhkan tingkat komputasi yang lebih tinggi. Hasil dari teknik ini adalah *subset* fitur yang memberikan kontribusi paling baik. *Feature subset* bisa dilakukan dengan memanfaatkan metode *sequential forward selection* (dari satu menjadi banyak fitur), *sequential backward selection* (dari banyak menjadi satu), *sequential floating selection* (bisa dari mana saja), *GA, Greedy Search, Hill Climbing, Simulated Annealing* dan lain-lain.

Teknik *embedded* merupakan penggabungan keunggulan dari teknik *filter* yang cepat dan akurasi dari teknik *wrapper*. Pada teknik *embedded*, seleksi fitur dilakukan sebagai bagian dari proses *learning* terhadap seluruh data *training*, sehingga pada umumnya hasil seleksi fitur spesifik ke model yang dibangun. Dalam teknik *embedded* ini, fitur secara natural dihilangkan, apabila *learning*

machine menganggap fitur tersebut tidak begitu berpengaruh. Beberapa *learning machine* yang bisa digunakan antara lain: *Decision Trees*, *Random Forests* dan lain-lain.

a. *Entropy*

Di dalam bidang *Information Theory*, *Entropy* sering digunakan sebagai suatu parameter untuk mengukur heterogenitas (keberagaman) dari suatu kumpulan sampel data. Jika kumpulan sampel data semakin heterogen, maka nilai *entropy* nya semakin besar. Secara matematis, *entropy* dirumuskan sebagai berikut :

$$Entropy(S) = \sum_{i}^{c} - \ p_i \ log_2 \ p_i \qquad (2.1)$$

Dimana :

c : Jumlah nilai yang ada pada atribut target (jumlah kelas klasifikasi)

p_i : Jumlah sampel untuk kelas *i*

b. *Information Gain*

Setelah mendapatkan nilai *entropy* untuk suatu kumpulan sampel data, maka dapat diukur efektivitas suatu fitur dalam mengklasifikasikan data. Ukuran efektivitas ini disebut sebagai *information gain*. Secara matematis, *information gain* dari suatu fitur *A*, dituliskan sebagai berikut (Suyanto, 2007) :

$$IG\ (S, A) = \ Entropy(S) \ - \sum_{v \equiv Values(A)} \frac{|S_v|}{S} * Entropy(S_v) \qquad (2.2)$$

Dimana :

A : fitur

V : menyatakan suatu nilai yang mungkin untuk fitur A

Values(*A*) : himpunan nilai-nilai yang mungkin untuk fitur A

$|S_v|$: jumlah sampel untuk nilai *v*

|S| : jumlah seluruh sampel data

Entropy(S_v) : *entropy* untuk sampel-sampel yang memiliki nilai *v*

c. *Symmetric Uncertainty*

SU mengkompensasi bias *IG* terhadap fitur dengan nilai lebih tersendiri dan menormalkan nilai-nilai berkisaran 0 hingga 1. Pengukuran *SU* dapat dihitung dengan persamaan sebagai berikut (Firqiani dkk, 2007) :

$$SU\ (S,A) = \ 2 * \frac{IG(S,A)}{H(S) + H(A)} \qquad (2.3)$$

Dimana :

A : fitur

S : kelas

H : nilai *entropy*

Nilai *Symmetrical Uncertainty* (SU) berkisar pada rentang 0 sampai dengan 1. Fitur akan terpilih jika nilai *SU*>δ, dimana δ adalah nilai *threshold* = 0. Pada penelitian Firqiani dkk, (2005) nilai akurasi tertinggi terdapat pada nilai *threshold* 0.

2.1.4.4 *K-Nearest Neighbor* (K-NN)

Zainuddin et al (2013) algoritma *k-nearest neighbor* (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut (sebagaimana dikutip oleh Maihendra, 2016). Sedangkan menurut Wu (2009), sebagaimana dikutip oleh Maihendra (2016), metode k-*Nearest Neighbors* merupakan metode klasifikasi dengan teknik *lazy learning*. Metode ini dilakukan dengan cara mencari kelompok *k* objek di dalam data *training* yang memiliki kemiripan paling dekat dengan data *testing*. Dengan kata lain, K-NN mengklasifikan data dengan perhitungan jarak terpendek sebagai ukuran dalam klasifikasi data-data baru.

Metode K-NN digolongkan dalam algoritma *supervised* yaitu proses pembentukan algoritma diperoleh melalui proses pembelajaran pada *record* lama yang telah terklasifikasi dan hasil pembelajaran tersebut digunakan untuk

mengklasifikasikan *record* baru dengan *output* yang belum diketahui. Algoritma *supervised* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola-pola yang sudah ada sebelumnya.

Metode K-NN berpatokan pada tingkat kemiripan yang memiliki jarak terdekat terhadap data pola. Jumlah data tetangga terdekat dinyatakan dalam *k* yang berarti sebagai nilai dari kedekatan data-data. Misalkan ditentukan k=3, maka kasus dengan 3 jarak terdekat dipilih lalu diklasifikasi berdasarkan *instance* kelas target dimana kasus dengan jumlah mayoritas *instance* kelas target ditentukan sebagai klasifikasi untuk kasus baru.

Rumus-rumus yang biasa digunakan sebagai ukuran jarak untuk data numerik ini antara lain :

a. *Euclidean Distance*

Proses perhitungan jarak *euclidean* dari algoritma MK-NN ini adalah digunakan untuk mencari jarak tiap data latih dan mencari jarak tiap data uji dan data latih. Berikut persamaan yang digunakan untuk memperoleh nilai Jarak *Euclidean* (Werdani, Ajeng Kesuma, 2015) :

$$d_i = \sqrt{\sum_{i=1}^{p}(x_{2i} - x_{1i})^2} \qquad (2.4)$$

Keterangan :

x_1 = Data latih

x_2 = Data uji

i = Variabel data

d = Jarak

p = Dimensi data

b. *City Block Distance*

Jika tiap item digambarkan sebagai sebuah titik dalam grid, ukuran jarak ini merupakan banyak sisi yang harus dilewati suatu titik untuk mencapai titik yang lain seperti halnya dalam sebuah peta jalan. Berikut persamaan yang digunakan untuk menghitung *City Block Distance* adalah :

$$d(x,y) = \sum_{i=1}^{n}(x_i - y_i) \qquad (2.5)$$

Dengan :

d : jarak antara titik pada data *training* x dan titik data *testing* y yang akan diklasifikasi, dimana x=x1,x2,…,xi dan y=y1,y2,…,yi

x : data uji

y : data latih

i : merepresentasikan nilai atribut

n : merupakan dimensi atribut

c. *Manhattan Distance*

Manhattan Distance merupakan salah satu pengukuran yang paling banyak digunakan meliputi penggantian perbedaan kuadrat dengan menjumlahkan perbedaan absolute dari variable-variable. Fungsi ini hanya akan menjumlahkan selisih nilai x dan y dari dua buah titik. Berikut persamaan yang digunakan untuk menghitung *City Block Distance* adalah :

$$d(x,y) = \sum_{i=1}^{n}(x_i - y_i) \qquad (2.6)$$

Dengan :

d : jarak antara titik pada data *training* x dan titik data *testing* y yang akan diklasifikasi, dimana x = x1,x2,…,xi dan y = y1,y2,…,yi

x : data uji

y : data latih

i : merepresentasikan nilai atribut

n : merupakan dimensi atribut.

d. Minkwoski Metric

Ukuran ini merupakan bentuk umum dari *Euclidean Distance* dan *Manhattan Distance.Euclidean Distance* adalah kasus dimana nilai p=2 sedangkan *Manhattan Distance* merupakan bentuk *Minkwoski* dengan p=1.

Dengan demikian, lebih banyak nilai numerik yang dapat ditempatkan pada jarak terjauh di antara 2 vektor. Seperti pada *Euclidean Distance* dan juga *Manhattan Distance*, ukuran ini memiliki masalah jika salah satu atribut dalam vektor memiliki rentang yang lebih besar dibandingkan atribut-atribut lainnya. Berikut *minkwoski Metric* dalam dua dimensi :

$$d(x,y) = \sqrt[k]{x^k + y^k} \qquad (2.7)$$

Dengan :

d : jarak antara titik pada data *training* x dan titik data *testing* y yang akan diklasifikasi, dimana x=x1,x2,…,xi dan y=y1,y2,…,yi

x : data uji

y : data latih

k : 2 gives Euclidean distance

k : 1 gives city-block distance

Sedangkan *Minkowski metric* dalam *p* dimensi adalah :

$$Distance_{i,h} = \sqrt[k]{\sum_{i=1}^{p} (a_{i,j} - a_{h,j})^k} \qquad (2.8)$$

Dengan :

k : 2 gives Euclidean distance

k : 1 gives city-block distance

d. *Cosine*

Ukuran ini bagus digunakan pada data dengan tingkat kemiripan tinggi walaupun sering pula digunakan bersama pendekatan lain untuk membatasi dimensi dari permasalahan. Dalam mendefenisikan ukuran jarak antar k yang digunakan beberapa algoritma untuk menentukan k mana yang terdekat.

Bramer (2007) mengemukakan secara umum, rumus *eucliden distance* lebih sering digunakan (sebagaimana dikutip oleh Maihendra, 2016).

2.1.4.5 *Modified K-Nearest Neighbor* (MK-NN)

Hamid Parvin dkk (2008), sebagaimana dikutip oleh Maihendra (2016), metode ini merupakan modifikasi dari kNN dimana ada beberapa perhitungan yang ditambah. Setelah mendapatkan *euclidean distance*, selanjutnya dilakukan perhitungan validitas untuk semua data yang terdapat pada data latih. Kemudian dilakukan perhitungan Weight Voting pada semua data uji menggunakan validitas data.

a. Validitas Data Latih

Validitas digunakan untuk menghitung jumlah titik dengan label yang sama untuk semua data pada data latih. Validitas setiap data tergantung pada setiap tetangga terdekatnya. Setelah dilakukan validasi data, selanjutnya data tersebut digunakan sebagai informasi lebih mengenai data tersebut. Persamaan yang digunakan untuk menghitung validitas setiap data latih adalah (Rasepta, 2016) :

$$Validasi\ x = \frac{1}{3}\sum_{i=1}^{k} S(label\ x, label\ N_i\ x) \qquad (2.9)$$

Dimana :

K : Jumlah titik terdekat

Lbl (*x*) : Kelas *x*

Ni (*x*) : Label kelas titik terdekat *x*

Fungsi S digunakan untuk menghitung kesamaan antara titik *a* dan data ke-*b* tetangga terdekat. Persamaan untuk mendefinisikan fungsi S terdapat dalam Persamaan (2.11) di bawah ini (Rasepta, 2016) :

$$S\ a, b = \begin{matrix} 1\ a = b \\ 0\ a \neq b \end{matrix} \qquad (2.10)$$

Dimana :

a : Kelas *a* pada data training

b : Kelas lain selain *a* pada data training

b. Weight Voting

Dalam metode MK-NN, pertama *weight* masing-masing tetangga dihitung dengan menggunakan 1 / (d_e + 1). Kemudian, validitas dari setiap data pada data latih dikalikan dengan weight berdasarkan pada jarak *Euclidean*. Sehingga metode MKNN, didapatkan persamaan weight voting tiap tetangga sebagai berikut (Rasepta, 2016) :

$$W(i) = Validitas(i) \text{ x} \frac{1}{de+0,5} \qquad (2.11)$$

Dimana :

$W(i)$: Perhitungan *Weight Voting*

Validasi (i) : Nilai Validasi

d_e : Jarak *Euclidean*

2.1.5 Evaluasi

Performa dari suatu model kasifikasi dapat diukur tingkat akurasinya dengan melakukan evaluasi. Menurut Rasepta, (2016) Performa dari suatu model kasifikasi dapat diukur dengan tingkat akurasinya berdasarkan *Confusion matrix*. *Confusion matrix* merupakan alat yang berguna untuk menganalisis seberapa baik classifier mengenali tuple dari kelas yang berbeda. TP dan TN memberikan informasi ketika classifier benar, sedangkan FP dan FN memberikan informasi ketika classifier salah. Gambar (2.1) adalah contoh dari *confusion matrix*.

		Actual Class	
		Ya	Tidak
Predictive Class	Ya	TP	FN
	Tidak	FP	TN
	Total	P’	N’

Gambar 2.2 Confusion Matrix (Rasepta, 2016)

Akurasi merupakan persentase dari data yang diprediksi secara benar. Perhitungan akurasi adalah :

$$Akurasi = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2.12)$$

Keterangan :

TP : *True positives*, merupakan jumlah data dengan kelas positif yang diklasifikasi kan positif.

TN : *True negatives*, merupakan jumlah data dengan kelas negatif yang diklasifika sikan negatif.

FP : *False positives*, merupakan jumlah data dengan kelas positif diklasifikasikan negatif.

FN : *False negatives*, merupakan jumlah data dengan kelas negatif diklasifikasikan positif.

2.2 KDD CUP

KDD CUP merupakan suatu kompetisi di bidang Data Mining dan Ekplorasi ilmu pengetahuan diseluruh dunia yang diadakan oleh *ACM SIGKDD* (*Special Interest Group on Knowledge Discovery and Data Mining*). Organisasi ini menyelengarakan kompetisi tersebut pada setiap tahun dengan fokus tema yang berbeda-beda. Pada tahun 1999 kompetisi KDD Cup berfokus kepada *Intrusion Detection and Report*. *Intrusion Detection and Report* merupakan suatu data laporan intrusi serangan pada jaringan komputer yang dapat digunakan sebagai acuan data latih dan uji untuk mendeteksi ancaman serangan (*www.kdnuggets.com*).

2.2.1 KDD CUP 1999 Dataset

Pada tahun 1999, ACM Special Interest Group on Knowledge Discovery and Data Mining adalah organisasi profesional terkemuka penambang data, menyelenggarakan kompetisi di dunia yang mempertemukan berbagai *researcher*, akademisi dan praktisi untuk dapat memberikan bantuan menyelesaikan kasus

yang diberikan dalam kompetisinya tersebut. Kompetisi tersebut adalah *Knowledge Discovery in Database* (KDD) Cup 99 yang bertema *Computer Network Intrusion Detection*. Dataset KDD CUP 99 dikeluarkan oleh DARPA (*Defense Anvanced Research Projects Agency*) dan dikelola oleh MIT Loncoln Lbs. Kumpulan dari data ini digunakan sebagai alat kompetisi internasional ilmu pengetahuan dan data mining yang ke 3, yang diadakan secara bersamaan dengan konfrensi internasional ilmu pengetahuan dan data mining KDD-99 yang ke lima, tujuan dari kompetisi adalah untuk membangun detektor intrusi jaringan, yang mampu membuat model perbedaan prediksi antara koneksi "buruk" disebut dengan gangguan atau serangan dan baik disebut koneksi normal (*http://kdd.ics.uci.edu*).

Dataset pelatihan KDD terdiri dari sekitar 4.900.000 vektor koneksi tunggal yang masing-masing berisi 41 fitur dan diberi label sebagai sebuah serangan atau normal, dengan tipe satu jenis serangan tertentu.

Adapun fitur dari dataset KDD CUP 1999 yaitu sebagai berikut :

Tabel 2.1 Fitur data KDD CUP 1999

No	Fitur	Deskripsi
1	*Duration*	Lama (detik) koneksi
2	*protocol_type*	Tipe protokol (tcp, udp, dll)
3	*service*	*Network service* di *destination* (http, telnet, dll)
4	*Flag*	*flag*
5	*scr_bytes*	Jumlah *byte*s dari *source* ke *destination*
6	*dst_bytes*	Jumlah *bytes* dari *destination* ke *source*

No	Fitur	Deskripsi
7	*Land*	Status koneksi, normal atau *error*
8	*wrong_fragment*	Jumlah *fragment* yang salah
9	*Urgent*	Jumlah paket yang *urgent*
10	*Count*	Jumlah koneksi ke *host* yang sama dengan koneksi yang ada sekarang dalam rentang 2 detik
11	*serror_rate*	% dari keneksi yang terdapat "SYN" *error*
12	*rerror_rate*	% dari koneksi yang terdapat "REJ" *error*
13	*same_srv_rate*	% dari koneksi ke *service* yang sama
14	*diff_srv_rate*	% dari koneksi ke *service* yang berbeda
15	*srv_count*	Jumlah koneksi ke *service* yang sama terakhir
16	*srv_serror_rate*	% dari koneksi yang terdapat "SYN"
17	*srv_rerror_rate*	*Error* % dari koneksi yang terdapat
18	*srv_diff_host_rate*	% dari koneksi ke *host* yang berbeda

No	Fitur	Deskripsi
19	*dst_host_count*	Jumlah koneksi ke *host* yang sama dengan koneksi ke *host* yang sama sekarang dalam rentang 2 detik
20	*dst_host_serror_rate*	% dari koneksi yang terdapat "SYN" *error*
21	*dst_host_rerror_rate*	% dari koneksi yang terdapat "REJ" *error*
22	*dst_host_same_srv_rate*	% dari koneksi ke *service* yang sama
23	*dst_host_diff_srv_rate*	% dari koneksi ke *service* yang berbeda
24	*dst_host_srv_count*	% dari koneksi yang terdapat "REJ"*error*
25	*dst_host_srv_serror_rate*	% dari koneksi yang terdapat "REJ"*error*
26	*dst_host_srv_rerror_rate*	% dari koneksi yang terdapat "REJ"*error*
27	*dst_host_srv_diff_host_rate*	% dari koneksi ke host yang berbeda
28	*dst_host_same_src_port_rate*	% dari koneksi ke *port service* yang sama
29	*Hot*	Jumlah indicator "*hot*" secara beruntun
30	*num_failed_logins*	Jumlah percobaan login yang gagal 1

No	Fitur	Deskripsi
31	*logged_in*	Jika berhasil login, o sebaliknya
32	*num_compromised*	Jumlah kondisi "*compromised*"
33	*root_sheel*	1 jika *root shell* didapat, 0 sebaliknya
34	*su_attempted*	1 jika dilakukan percobaan perintah "*su root*", 0 sebaliknya
35	*num_root*	Jumlah "*root*" yang diakses
36	*num_file_creations*	Jumlah operasi pembuaan file
37	*num_shells*	Jumlah *promt shell*
38	*num_access_files*	Jumlah operasi pada *access control files*
39	*num_outbound_cmds*	Berurutan
40	*is_host_login*	1 jika login termasuk dalam daftar "*hot*" 0 sebaliknya
41	*is_guest_login*	1 jika *login* adalah "*guest*", 0 sebaliknya
42	*Class*	Jenis trafik (normal, atau jenis seramgan tertentu)

Dataset ini merupakan data rekam koneksi yang terdiri dari 1 jenis data normal dan 22 jenis data serangan yang dikelompokkan kedalam empat tipe intrusi. Dataset yang memiliki 41 atribut/fitur yang dibagi ke dalam tiga kelompok yaitu *fitur basic*, *fitur konten* dan *fitur trafik*. *Fitur basic* (fitur nomor 1 sampai 9) merupakan hasil ekstraksi dari sistem *log tcpdump* dalam jaringan komputer. *Fitur konten* (fitur nomor 10 sampai 22) merupakan fitur-fitur yang diambil dari kegiatan yang berlangsung dalam sistem jaringan komputer. Sedangkan *fitur trafik* terbagi menjadi dua bagian, pertama terdiri dari fitur nomor 23 sampai 31 merupakan fitur trafik jaringan yang dihitung menggunakan waktu dua detik time window, dan kedua terdiri dari fitur nomor 32 sampai 41 dihitung menggunakan waktu dua detik time window dari tujuan ke host (Essra dkk, 2016).

Terdapat empat jenis/tipe kategori serangan-serangan simulasi sebagai berikut (Tavallaee, Mahbod dkk, 2009):

1. *Denial of Service Attack* (DoS) : adalah tipe serangan yang membebani sumber daya komputer (misalnya dengan *synflood* atau *ping of death*) sehingga komputer target mengalami sistem *crash* dan tidak mampu untuk memproses koneksi normal bahkan mengakibatkan user tidak dapat mengakses komputer tersebut.
2. *User to Root Attack* (U2R) : adalah tipe serangan yang berusaha untuk mendapatkan akses root atau admin pada komputer target dengan melakukan eksploitasi celah keamanan sistem. Serangan U2R umumnya dilakukan setelah penyerang mendapatkan akses user normal ke sistem (baik melalui sniffing, social engineering, ataupun dictionary attack).
3. *Remote to Local Attack* (R2L) : adalah tipe serangan yang bertujuan untuk mendapatkan akses sebagai pengguna sistem. R2L dilakukan oleh penyerang yang memiliki akses ke sistem dan melakukan eksploitasi untuk mendapatkan akses lokal.
4. *Probing Attack* : adalah bertujuan untuk mendapatkan informasi tentang status jaringan komputer dengan cara melakukan pemindaian terhadap komputer- komputer dalam jaringan tersebut. Informasi ini dapat

digunakan oleh penyerang untuk memetakan jaringan yang berguna dalam melakukan penyerangan berikutnya.

Tabel 2.2 Klasifikasi Tipe Serangan Jaringan (Ginting, 2016)

Tipe Serangan	Kelas
normal	Normal
apache2, back, land,mailbomb, neptune, pod, processtable, smurf, teardrop, udpstorm	DOS
butter_overflow, loadmodule, perl, ps, rootkit, sqlattack, xterm	U2R
ftp_write, guess_passwd, sendmail, imap, multihop, named, phf, snmpgetattack, snmpguess, warezmaster, worm, xlock, httptunnel, xznoop, wazerclient, spy	R2L
ipsweep, mscan, portsweep, saint, satan, nmap	Probe

Adapun didalam pelatihan data set, ada 23 jenis serangan yang muncul yang dikelompokkan kedalam 5 kelas seperti tabel 2.11 di bawah ini yakni normal, R2L, U2R, Probe dan DoS.

Tabel 2.3 Class Labels and the Number of Samples that Appears in "10%" KDD" Dataset (Rampure dan Tiwari, 2015)

Attack	Original Number of Samples	Class Level
back	2203	DOS
land	21	DOS
neptune	107201	DOS
pod	264	DOS
smurf	280790	DOS
teadrop	979	DOS
satan	1589	PROBE

Attack	Original Number of Samples	Class Level
ipsweep	1247	PROBE
nmap	231	PROBE
portsweep	1040	PROBE
normal	97277	NORMAL
guess_passwd	53	R2L
ftp_write	8	R2L
imap	12	R2L
phf	4	R2L
multihop	7	R2L
warzemaster	20	R2L
warzclient	1020	R2L
spy	2	R2L
buffer_overflow	30	U2R
loadmodule	9	U2R
perl	3	U2R
rootkit	10	U2R

2.3 Penelitian Terkait

Tabel 2.4 Penelitian Terkait

Penulis	Judul penelitian	Kesimpulan
Fakihatin Wafiyah, Nurul Hidayat dan Rizal Setya Perdana (2017)	*Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam*	Dari uji coba yang dilakukan penelitian ini mempelajari pola dari data hasil pemeriksaan diagnosa sementara dengan menerapkan metode klasifikasi yaitu *Modified K-Nearest Neighbor (MKNN)* berdasarkan 15 gejala penyakit demam dengan proses perhitungan jarak

Penulis	Judul penelitian	Kesimpulan
		eucledian, perhitungan nilai validitas dan perhitungan *weighted voting* yang hasil akhirnya digunakan untuk penetapan kelas klasifikasi berdasarkan nilai *K* yang telah ditentukan. Berdasarkan hasil pengujian terhadap perubahan nilai *K*, perubahan jumlah data latih dan perubahan komposisi data latih didapat kan rata-rata akurasi untuk pengujian pengaruh nilai *K* terhadap akurasi sebesar 88.55%. Nilai rata-rata akurasi yang didapatkan dari pengujian pengaruh variasi jumlah data latih adalah 92.42%. Pengujian pengaruh komposisi data latih terhadap akurasi mendapatkan nilai rata-rata akurasi sebesar 87.89%. Pengujian pengaruh komposisi data latih dan data uji terhadap akurasi mendapatkan nilai rata-rata akurasi sebesar 96.35%.
Naldi Nurhadi (2017)	*Aplikasi Intelligence Intrusion Detection System (IIDS) Dengan Menggunakan Metode K-Nearest Neighbor Untuk Mendeteksi Serangan Pada Jaringan*	Pada penelitian ini untuk mengklasifikasi kan serangan pada jaringan menggunakan Algoritma k-Nearest Neighbor. Pengujian sistem klasifikasi serangan pada jaringan dilakukan dengan confusion matrix. Pengujian untuk mengklasifikasikan serangan pada jaringan mendapatkan hasil yang baik. Nilai akurasi pengujian dari seluruh dataset yang digunakan yaitu mencapai k3 = 80 % dan k13 = 85 %.

Penulis	Judul penelitian	Kesimpulan
Tri Halomoan Simanjuntak, Wayan Firdaus Mahmudy dan Sutrisno (2017)	*Implementasi Modified K-Nearest Neighbor Dengan Otomatisasi Nilai K Pada Pengklasifikasian Penyakit Tanaman Kedelai*	Penelitian terkait ini mengklasifikasikan jenis penyakit yang menyerang tanaman kedelai. Penelitian ini menggunakan *Soybean Disease Data Set* yang terdiri dari 266 data latih dan akan dibangun aplikasi berbasis desktop dengan mengimplementasikan metode algoritma *Modified K-Nearest Neighbor*, parameter nilai *K* ditentukan oleh sistem dengan menggunakan metode Brute Force sehingga menemukan nilai *K* terbaik. Setiap nilai *K* dengan akurasi hasil terbaik akan disimpan dan digunakan sebagai parameter nilai *K* pada proses pengujian data baru. Nilai *K* pada metode ini mendefinisikan jumlah tetangga terdekat yang digunakan untuk proses klasifikasi. Hasil pengujian menunjukkan bahwa parameter nilai *K* sangat berpengaruh terhadap hasil klasifikasi dan akurasi yang dihasilkan. Rata-rata akurasi cenderung menurun seiring dengan penambahan nilai k sedangkan peningkatan jumlah data latih turut disertai dengan peningkatan hasil akurasi, untuk data latih dengan kelas tidak seimbang mengalami penurunan nilai akurasi seiring dengan bertambahnya jumlah data. Hasil akurasi tertinggi pada pengujian ini sebesar 100% dengan nilai

Penulis	Judul penelitian	Kesimpulan
		k=1 dan rata-rata akurasi dari 5 percobaan sebesar 98,83%.
Ridho Maihendra (2016)	*Penerapan Meto de Modified K-Nearest (MK-NN) Neighbor untuk Mempredik si Putusan Perka ra Perceraian*	Penelitian ini melakukan penggalian informasi dan pola baru berdasarkan dari total 1200 data untuk memprediksi putusan perkara perceraian dengan menggunakan metode klasifikasi MK-NN. Terlebih dahulu dilakukan *preprocessing* sebelum dilakukan proses perhitungan dengan metode yang digunakan. Hasilnya sistem yang dibangun sesuai harapan dan mampu membantu calon pemohon atau penggugat cerai menghadapi perkara perceraiannya dengan persentase tingkat akurasi sebesar 95,089% pada perbandingan data latih dan data uji 80:20 dengan nilai k=3
Indera Zainul Mutaqien (2016)	*Pengembangan Metode Seleksi Fitur dan Transfo rmasi Data Pada Sistem Deteksi Int rusi Dengan Pem batasan Ukuran Cluster dan Sub-Medoid*	Pada penelitian ini diajukan suatu sistem deteksi intrusi yang terdiri dari serangkai an proses seleksi fitur, clustering, dan transformasi data dengan pendekatan metode centroid based. Proses seleksi fitur dilakukan secara bertahap dengan menggabungkan teknik filter dan wrapper untuk memperoleh fitur-fitur yang tepat. Sistem ini juga menggunakan nilai yang

Penulis	Judul penelitian	Kesimpulan
		disebut sebagai ambang radius untuk membatasi ukuran cluster yang terbentuk pada proses clustering. Proses transfor masi data dilakukan dengan memanfaat kan jarak data ke centroid dan jarak data ke beberapa sub-medoid untuk meningkat kan akurasi hasil deteksi. Hasil penelitian ini menunjukkan bahwa pemilihan fitur-fitur yang tepat (signifikan) pada dataset dapat meningkatkan performa hasil deteksi. Pada dataset NSLKDD yang digunakan pada penelitian ini ditemukan ada 19 fitur signifikan, sedangkan pada dataset Kyoto2006+ terdapat 14 fitur signifikan. Selain itu metode yang diajukan secara umum memberikan perbaikan hasil deteksi pada setiap dataset yang diuji. Hasil terbaik terlihat pada penerapan metode yang diajukan pada dataset Kyoto2006+.
Kevin Martha Rasepta (2016)	*Klasifikasi Status Gizi Balita Meng gunakan Metode Modified K-Nea rest Neighbor*	Penelitian terkait ini melakukan penggalian informasi dan pola baru berdasarkan dari total 400 data, 397 data latih dan 3 data uji untuk mengklasifikasi status gizi balita. Proses data selection untuk model klasifikasi akan dilakukan secara manual. Kemudian dilakukan proses transformasi, perhitungan jarak menggunakan Manhattan. Proses selanjut

Penulis	Judul penelitian	Kesimpulan
		nya menghitung validitas, dan weight voting. Hasil dari analisa perancangan model klasifikasi akan diimplementasikan pada sistem berbasis web. Sistem yang dibangun berdasarkan model klasifikasi tersebut diuji menggunakan Evaluasi, BlackBox, User Accepetence Test. Hasil nya sistem yang dibangun sesuai harapan dan mampu membantu calon user mengklasifikasi status gizi balita dengan tingkat akurasi tertinggi sebesar 90% pada skenario 90:10 dan 80:20 dengan k=1 sampai k=3, rata-rata tingkat akurasi adalah 82,057%.
Muhammad Arifin (2015)	*IG-KNN untuk Prediksi Costumer Churn Telekomunikasi*	Penelitian ini menggabungkan algoritma pemilihan fitur *information gain* dengan algoritma klasifikasi KNN untuk dapat meningkatkan akurasi dalam mem-prediksi *costumer churn* telekomunikasi, berdasarkan hasil penelitian ini dengan menggunakan IG-KNN menunjukkan akurasi lebih baik meski dengan nilai k yang berbeda-beda bila dibandingkan dengan menggunakan KNN tanpa fitur seleksi, adapun peningkatan akurasi yang dicapai dari k1 sampai k11 sebesar 1,7%
Siti Mutrofin, Abidatul Izzah, Arrie Kurnia	*Optimasi Teknik Klasifikasi Modi fied K Nearest Ne*	Penelitian ini melakukan perbaikan dalam hal kekurangan pada metode KNN, kekurangan KNN adalah nilai k bias,

Penulis	Judul penelitian	Kesimpulan
wardhani dan Mukhamad Masrur (2014)	*ighbor Menggunakan Algoritma Genetika*	komputasi kompleks, keterbatasan memori, dan mudah tertipu dengan atribut yang tidak relevan. Salah satu perbaikan kNN adalah Modified kNN (MKNN), yang bertujuan untuk meningkatkan akurasi dari kNN, dengan menambahkan perhitu ngan validity, karena dianggap perhitung an bobot yang terdapat pada kNN, memiliki permasalahan outlier. Namun, MKNN juga memiliki kelemahan yang sama dengan kNN yaitu nilai k bias dan komputasi yang kompleks. Berdasarkan permasalahan MKNN tersebut, penelitian terkait ini akan melakukan perbaikan dalam hal, optimasi nilai k menggunakan Genentic Algorithm (GA), karena GA sudah terbukti dapat digunakan untuk melakukan optimasi pada nilai k untuk kNN. Selanjutnya algoritma tersebut akan dinamakan algoritma GMKNN (Genetic Modied k Nearest Neighbor). Evaluasi tingkat kebenaran hasil akan didasarkan pada nilai akurasi, baik menggunakan algoritma kNN, MKNN maupun GMKNN menggunakan data UCI machine learning.
Bekti Maryuni Susanto (2014)	*K-Nearest Neighbor (KNN) untuk Mendeteksi Gangguan*	Berdasarkan hasil penelitian dapat disimpulkan bahwa algoritma k-NN dengan nilai k=1 memiliki tingkat akurasi terbesar dalam mendeteksi gangguan

Penulis	Judul penelitian	Kesimpulan
	Jaringan Komputer pada Intrusion Detection Dataset	jaringan komputer. Dari hasil eksperimen diperoleh tingkat akurasi algoritma k-nearest neigbour dalam mendeteksi gangguan jaringan komputer sebesar 79,36%. Banyaknya atribut yang digunakan pada penelitian ini membuat waktu yang dibutuhkan dalam melakukan testing cukup lama, sehingga penelitian selanjutnya bisa melakukan seleksi atribut yang relevan untuk mendeteksi gangguan jaringan komputer.
Hamid Parvin, Hosein Alizadeh dan Behrouz Minaeibidgoli (2008)	*MKNN : Modified K-Nearest Neighbor*	Dalam penelitian ini, algoritma baru untuk memperbaiki Kinerja pengklasifi kasi KNN yaitu Modified K-Nearest (MKNN). Metode yang meningkatkan kinerja Metode KNN menggunakan semacam preprocessing pada data latih. Hal tersebut menambah nilai baru bernama "Validity" untuk melatih sampel yang mana Menyebabkan lebih banyak informasi tentang situasi data pelatihan sampel di ruang fitur Validitas memper hitungkan akun Nilai stabilitas dan ketahanan dari setiap sampel latih Berkenaan dengan jarak (distance) nya. Penelitian menunjukkan MKNN sangat baik dalam peningkatan akurasi dibandingkan dengan metode KNN.

Penulis	Judul penelitian	Kesimpulan
Hida Nur Firqiani, Aziz Kustiyo dan Endang Purnama Giri (2007)	*Seleksi Fitur Menggunakan Fast Correlation Based Filter Pada Algoritma Voting Feature Intervals 5*	Hasil dari penelitian ini berupa perbandingan nilai akurasi data pada klasifikasi menggunakan Voting Feature Intervals 5 jika sebelumnya dilakukan seleksi fitur dan tidak dilakukan seleksi fitur. Hasil yang diperoleh menunjukkan bahwa nilai akurasi klasifikasi dengan seleksi fitur lebih baik dari pada tanpa seleksi fitur. Rata-rata hasil akurasi data tanpa seleksi fitur yaitu 81.66% sedangkan menggunakan seleksi fitur yaitu 85.51% .
Te-Shun Chou, Kang K. Yen, Jun Luo, Niki Pissinou dan Kia Makki (2007)	*Correlation-Based Feature Selection for Intrusion Detection Design*	Pada penelitian ini membandingkan pendekatan dua algoritma pemilihan fitur berbasis korelasi yakni *Correlation-based Filter Selection* (CFS) dan *Fast Correlation-Based Filter* (FCBF) di enam kumpulan data yang diambil dari UCI databases dan KDD99 dataset untuk melatih dan menguji algoritma pembelajaran mesin (*machine learning*) *C4.5* dan *Naive Bayes*. Dari hasil penelitian tersebut menunjukkan akurasi mencapai rata-rata tertinggi, memiliki kinerja superior dan dapat menseleksi fitur yang paling signifikan dalam memilih fitur informatif dari sekumpulan dataset untuk meningkatkan akurasi tugas klasifikasi.

Penulis	Judul penelitian	Kesimpulan
H. Günes Kayacik, A. Nur Zincir- Heywood dan Malcolm I. Heywood (2005)	*Selecting Features for Intrusion Detection : A Feature Relevance Analysis on KDD 99 Intrusion Detection Dataset*	Kontribusi dari penelitian ini menganalisa keterlibatan masing-masing fitur untuk klasifikasi. Hasilnya menunjukkan bahwa indikasi kelas normal, neptune dan smurf sangat berkaitan dengan fitur tertentu yang membuat klasifikasi menjadi lebih mudah. Karena ketiga kelas ini merupakan 98% dari data latih, sangat mudah bagi algoritma pembelajaran mesin untuk mencapai hasil yang baik tersebut. Di sisi lain, beberapa fitur tertentu tidak memiliki kontribusi terhadap deteksi intrusi, yang mengindika sikan bahwa tidak semua fitur berguna. Meskipun data uji menunjukkan karakte ristik yang berbeda dari pada data latih, karena "10% KDD" adalah data pelatihan dalam kompetisi, analisa data latih menjelaskan kinerja sistem deteksi intrusi berbasis mesin yang dilatih pada dataset.