

BAB II

LANDASAN TEORI

2.1 Penelitian Sebelumnya

Rufaidha et al. (2016) dengan judul Prediksi Penyakit Menggunakan Algoritma *K-Nearest Neighbor* Dan Algoritma Genetika Untuk Data Berdimensi Tinggi. Metode yang digunakan dalam membangun sistem informasi ini adalah metode *Euclidean distance*. Sistem yang dibangun digunakan untuk mengetahui performansi dari algoritma *K-NN* dalam memprediksi penyakit dengan data berdimensi tinggi.

Kurnianingtyas et al. (2017) dengan judul Sistem Pendukung Keputusan Diagnosis Penyakit Sapi Potong Menggunakan *K- Nearest Neighbour (K- NN)*. Metode yang digunakan dalam membangun sistem informasi ini adalah metode *K-Nearest Neighbour*. Sistem yang dibangun untuk membantu mempercepat penganalisan jenis penyakit pada hewan ternak dengan memberikan informasi terkait jenis penyakit yang menjangkit hewan ternak dan solusi penanganannya.

Triatmaja, Hidayat, and Mahfud (2018) tentang Sistem Pendukung Keputusan Diagnosis Penyakit Sapi Potong Menggunakan *K- Nearest Neighbour (K- Nn)*. Metode yang digunakan dalam membangun sistem informasi ini adalah Metode *Neighbor Weighted K-Nearest Neighbor*. Dengan sistem tersebut, dapat membantu para peternak dalam mendiagnosis penyakit sapi berdasarkan pengetahuan pakar yang dikombinasikan dengan

algoritma NWK-NN. Keluaran yang dihasilkan dari pengolahan sistem adalah jenis penyakit yang sedang menyerang sapi.

Satya and Hidayat (2018) Sistem Pakar Diagnosis Penyakit Sapi Ternak Potong Menggunakan Metode *Naïve Bayes - Certainty Factor*. Metode yang digunakan dalam membangun sistem informasi ini adalah Metode *Naïve Bayes - Certainty Factor*. Sistem ini dapat mendiagnosa penyakit pada hewan ternak sapi yang mudah dimengerti dan dapat diakses oleh semua kalangan masyarakat lewat internet.

Perbandingan penelitian dapat dilihat pada table berikut.

Tabel 2.1 Perbandingan Penelitian

No.	Nama	Judul Penelitian	Metode	Hasil
1.	Rufaidha et al. (2016)	Prediksi Penyakit Menggunakan Algoritma <i>K-Nearest Neighbour</i> Dan Algoritma Genetika Untuk Data Berdimensi Tinggi.	<i>Euclidean distance</i>	Sistem ini dapat digunakan untuk mengetahui performansi dari algoritma K-NN dalam memprediksi penyakit dengan data berdimensi tinggi.

2	Kurnianingtyas et al. (2017)	Sistem Pendukung Keputusan Diagnosis Penyakit Sapi Potong Menggunakan <i>K-Nearest Neighbour</i> (<i>K- NN</i>).	<i>K-Nearest Neighbour</i>	Sistem ini digunakan untuk membantu mempercepat penganalisisan jenis penyakit pada hewan ternak dengan memberikan informasi terkait jenis penyakit yang menjangkit hewan ternak dan solusi penanganannya
3	Triatmaja, Hidayat, and Mahfud (2018)	Sistem Pendukung Keputusan Diagnosis Penyakit Sapi Potong Menggunakan <i>K-Nearest Neighbour</i>	<i>K- Nearest Neighbour</i> (<i>K- Nn</i>)	Dengan sistem tersebut, dapat membantu para peternak dalam mendiagnosis penyakit sapi berdasarkan pengetahuan pakar yang dikombinasikan dengan algoritma NWK-NN. Keluaran yang dihasilkan dari pengolahan sistem adalah jenis penyakit yang sedang menyerang sapi.
4	Satya and Hidayat (2018)	Sistem Pakar Diagnosis Penyakit Sapi Ternak	<i>Naïve Bayes - Certainty</i>	Sistem ini dapat mendiagnosa

		Potong Menggunakan Metode <i>Naïve</i> <i>Bayes - Certainty</i> <i>Factor</i>	<i>Factor</i>	penyakit pada hewan ternak sapi yang mudah dimengerti dan dapat diakses oleh semua kalangan masyarakat lewat internet.
--	--	---	---------------	--

Berdasarkan uraian penelitian terdahulu di atas perbandingannya dengan penelitian ini yaitu sebagai berikut:

Penelitian yang dilakukan Kurnianingtyas et al. (2017) juga menggunakan metode *K-NN* dan menggunakan pengujian akurasi serta data gejala yang digunakan berjumlah 20 data gejala, sedangkan penelitian yang dilakukan ini lebih khusus berfokus pada perhitungan dengan metode *k-nearest neighbor* saja dan data gejala yang akan digunakan berjumlah 24 data gejala.

Penelitian yang dilakukan oleh Triatmaja, Hidayat, and Mahfud (2018) menggunakan metode Neighbor Weighted *K-Nearest Neighbor* (NWK-NN), dimana yang membedakan antara metode NWK-NN dengan metode *K-NN* pada penelitian ini adalah adanya pembobotan untuk setiap jenis/kelas dan proses perhitungan skor untuk menentukan klasifikasi terhadap data uji. pemberian bobot pada kelas/jenis yang berasal dari kategori mayoritas akan diberi nilai bobot kecil, sedangkan pada kategori minoritas akan diberi nilai bobot besar.

Penelitian yang dilakukan oleh Satya and Hidayat (2018) menggunakan metode Naïve Bayes - Certainty Factor dimana proses perhitungannya dengan cara menghitung peluang probabilitas penyakit. Proses mencari peluang semua penyakit dibagi seluruh jumlah penyakit, lalu selanjutnya proses *Likelihood* yang dimana menghitung setiap gejala pada masing masing penyakit yang ada, kemudian langkah terakhir dari *Naïve Bayes* adalah perhitungan Posterior yang dimana mencari nilai tertinggi hasil *Likelihood* dikalikan dengan hasil *Prior* dengan mencari nilai yang tertinggi, yang dimana nilai tertinggi itu akan menjadi hasil dari diagnosis sedangkan pada penelitian ini menggunakan metode *K-NN* yaitu dengan cara menentukan parameter *k*, menghitung jarak euclidean, mengurutkan hasil jarak euclidean dan mengumpulkan kategori mayoritas sehingga dapat di prediksi kategori objek atau kelasnya.

Penelitian yang dilakukan oleh Rufaidha et al. (2016) menggunakan kasus penyakit pada manusia sedangkan pada penelitian ini menggunakan kasus penyakit pada sapi jenis sapi bali. Algoritma yang di gunakan yaitu *k-nearest neighbor* dan algoritma genetika sedangkan penelitian ini hanya menggunakan algoritma *k-nearest neighbor*. Penelitian ini akan menghasilkan sebuah sistem yang memiliki menu utama yakni Klasifikasi Penyakit. Menu Klasifikasi Penyakit akan digunakan oleh masyarakat untuk melakukan klasifikasi penyakit berdasarkan gejala yang muncul pada sapi sehingga nantinya sistem akan mengeluarkan hasil berupa jenis penyakit yang cocok sesuai gejala yang di input oleh masyarakat (user), serta cara penanganan

awal dari penyakit tersebut sehingga dapat dimanfaatkan oleh masyarakat untuk memperoleh informasi mengenai cara menangani lebih awal yang telah disediakan oleh sistem ini. Metode yang digunakan dalam penelitian ini adalah metode *waterfall* dengan bantuan perangkat lunak pendukung pembuatan Sistem ini yakni *Sublime Text*, *HTML*, *CSS*, *PHP*, *Phpmyadmin*, dan *MySQL*.

2.2 Penyakit Sapi Bali

Berdasarkan data yang diambil dari Dinas Peternakan NTT, ada 6 penyakit yang dapat diderita oleh ternak sapi bali. Berikut adalah penyakit sapi bali yang digunakan dalam penelitian ini.

a. Penyakit Jembrana / Keringat darah

Penyakit keringat darah (Penyakit Jembrana) ini disebabkan oleh virus jembrana (*retrovirus*) dan bersifat ganas pada sapi bali. Penularan penyakit jembrana dari sapi ke sapi lainnya diperkirakan oleh serangga penghisap darah seperti lalat (*lalat tapis*) caplak dan nyamuk. Gejala penyakit yaitu demam (suhu badan tinggi), bengkak pada kelenjar limfe, luka-luka (erosi) pada selaput lendir mulut, nafsu makan menurun, diare yang sering bercampur darah dan juga sering berkeringat darah.

b. Penyakit BEF (*Bovine Ephemeral Fever*)

Penyakit BEF pada ternak sapi disebabkan oleh virus (*Double Stranded Ribonucleic Acid*). Penyakit ini ditularkan oleh serangga (*arthropod borne viral disease*), bersifat benign non contagious, yang

ditandai dengan demam mendadak, sapi tidak mau makan dan kaku pada persendian. Penyakit dapat sembuh kembali beberapa hari kemudian.

c. Penyakit Bali Ziekte

Penyakit bali ziekte yaitu kondisi gangguan kesehatan ternak sapi akibat keracunan zat yang terdapat pada tanaman Tembelekan/Saliara (*Lantana camara*) atau kerasi. Gejala penyakit bali/bali ziekte yaitu nafsu makan menurun, suhu badan sapi tinggi, gatal-gatal dan sapi tidak tenang, kulit sapi dibagian tubuh yang menonjol dan ujung telinga mengering dan mengelupas.

d. Penyakit Diare Ganas Sapi (DGS) atau Bovine Viral Diarrhea (BVD)

Bovine Viral Diarrhea (BVD) atau *Diare Ganas Sapi* (DGS) adalah penyakit hewan menular yang akut dan sering berakibat fatal, disebabkan oleh virus dari genus *Pestivirus* dari famili *Togaviridae* (Dharma dan Putra, 1997). Umumnya infeksi paska kelahiran bersifat non klinis, peningkatan temperatur biphasic (terjadi dua kali peningkatan suhu badan) dan leukopenia yang diikuti peningkatan zat kebal/antibodi yang dapat dideteksi dengan uji serum netralisasi. Infeksi dapat dilihat melalui diagnosis serologik, virologik dan munculnya tanda klinis serta adanya lesi patologik. Gejala penyakit Diare Ganas Sapi (DGS) atau Bovine Viral Diarrhea (BVD) yaitu nafsu makan menurun, hewan tampak lesu, diare dengan kotoran sangat cair dan berbau busuk, suhu badan tinggi serta gangguan pernapasan.

e. Penyakit Berak Darah (*Coccidiosis*)

Penyebab penyakit berak darah (*Coccidiosis*) pada ternak sapi disebabkan oleh bakteri *E. bovis*, *E. zurni*, *E. ellipsoidalis* dan *E. auburnensis*. Gejala klinis penyakit berak darah bervariasi, tergantung spesies bakteri penyerang dan umur sapi. Sapi yang berumur dibawah 6 bulan sangat peka terhadap penyakit ini. Pada infeksi ringan ditandai oleh adanya diare ringan yang berlangsung sekitar 5-7 hari, pada infeksi berat tinja sering terlihat bercampur darah dan lendir, hewan penderita akan mengalami depresi, nafsu makan menurun sampai hilang, berat badan menurun, dehidrasi dan daerah sekitar anus menjadi kotor oleh adanya tinja

f. Penyakit Cacingan pada Sapi

Cacingan merupakan penyakit yang sering dialami oleh ternak sapi. Meskipun cacingan tergolong penyakit yang mudah ditangani namun penanganan harus dilakukan segera untuk menghindari kerugian akibat penyakit ini. Cacing yang menyerang ternak sapi sangat banyak jenisnya, tetapi yang paling sering menyerang adalah cacing pita dan cacing hati. Cacingan pada sapi disebabkan oleh kondisi pakan yang tidak bersih, biasanya pada rumput hujauan yang mengandung larva cacing. Gejala penyakit cacingan pada sapi yaitu tidak nafsu makan, terlihat kurus, susah buang air besar, diare, ternak tidak segar/ tampak lesu, nafas tidak beraturan, hidung dan mulut kering.

2.2.1 Tabel data gejala

Tabel data gejala dibawah ini diberi kode gejala agar dapat menghasilkan *rule* yang menghubungkan data penyakit dan data gejala

Tabel 2.2 Data Gejala

Kode Gejala	Nama Gejala
G1	Bengkak pada kelenjar limfe
G2	Berat badan menurun
G3	Daerah sekitar anus menjadi kotor oleh adanya
G4	Dehidarasi
G5	Demam (suhu badan tinggi)
G6	Depresi,
G7	Diare dengan kotoran sangat cair dan berbau busuk
G8	Diare ringan yang berlangsung sekitar 5-7 hari
G9	Diare yang sering bercampur darah
G10	Gangguan pernapasan
G11	Gatal-gatal
G12	Hewan tampak lesu
G13	Hidung dan mulut kering.
G14	Kaku pada persendian.
G15	Keringat darah.
G16	Kulit sapi dibagian tubuh yang menonjol dan ujung telinga mengering dan mengelupas
G17	Luka-luka (erosi) pada selaput lendir mulut
G18	Nafas tidak beraturan
G19	Nafsu makan menurun
G20	Sapi tidak mau makan
G21	Sapi tidak tenang

G22	Susah buang air besar
G23	Terlihat kurus
G24	Tinja sering terlihat bercampur darah dan lendir

2.3 Sistem

Sistem adalah suatu kesatuan, baik obyek nyata atau abstrak yang terdiri dari berbagai komponen atau unsur yang saling berkaitan, saling tergantung, saling mendukung, dan secara keseluruhan bersatu dalam satu kesatuan untuk mencapai tujuan tertentu secara efektif dan efisien.

Menurut Romney and Steinbart (2015) Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

2.4 Web

World Wide Web (WWW) adalah sistem pengakses informasi dalam internet yang biasa dikenal dengan istilah *web*. *Web* menggunakan *protokol* yang disebut *HTTP (HyperText Transfer Protocol)* yang berjalan pada *TCP/IP*. Dengan menggunakan *HyperText*, pemakai dapat melompat dari suatu dokumen ke dokumen lain dengan mudah, dengan cukup mengklik *text-text* khusus yang pada awalnya ditandai dengan garis bawah. Penggunaan *HyperText* pada *web* juga telah dikembangkan lebih jauh menjadi *HyperMedia*. Dengan menggunakan pendekatan *HyperMedia*, tidak hanya *text* yang dapat dikaitkan, melainkan juga gambar, suara, dan bahkan video (Kadir, 2014)

2.5 Pengertian Desain Sistem

Tahap setelah *analisis* sistem dari siklus pengembangan sistem yang mendefinisikan dari kebutuhan-kebutuhan fungsional, persiapan untuk rancang bangun, *implementasi* menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi (Hartono_Musakini 2014).

2.6 Perancangan sistem

Percancangan sistem terdiri dari antarmuka perancangan, *flowchart*, relasi dan *Entity Relationship Diagram* (E-R Diagram).

2.6.1 Antarmuka perancangan

Manfaat perancangan antarmuka adalah agar pengguna dan komputer dapat saling berinteraksi, sehingga pengguna merasakan adanya kemudahan dan keramahan sistem komputer kepadanya, diperlukan suatu media yang memungkinkan interaksi tersebut secara langsung. Perangkat lunak (Software) yang diperlukan dalam perancangan dan pembuatan aplikasi ini adalah sebagai berikut :

1. *Flowchart* (Bagan alir)

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu *program*. *Flowchart* biasanya mempermudah penyelesaian suatu masalah, khususnya masalah yang dipelajari dan dievaluasi lebih lanjut. Bagan alir merupakan bagan yang menunjukkan arus kerja atau apa yang dikerjakan didalam sistem

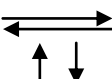
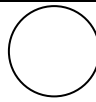
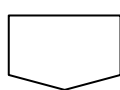
secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada dalam sistem. Pada bagian ini akan digambarkan *flowchart* sistem yang akan dibangun. *User* memasukan data kemudian disimpan (proses rekam) kedalam *database* dan juga mengalami proses rekam pada *database*. Tujuan adanya *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, dengan menggunakan simbol-simbol standar (Jogiyanto, 2010)

Simbol-simbol yang ada pada *flowchart* terbagi atas beberapa bagian yaitu :

a. *Flow direction symbol*

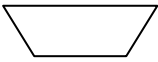
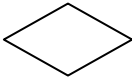
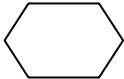
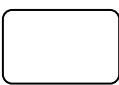
Digunakan untuk menghubungkan simbol satu dengan yang lain.

Tabel 2.3 Simbol *flow direction*

No	Simbol	Keterangan
1.		Menyatakan jalannya arus suatu proses.
2.		Menyatakan transmisi data dari satu lokasi ke lokasi lain.
3.		Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
4.		Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.

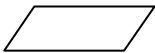
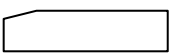
Processing symbol, menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur.

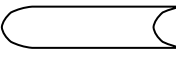
Tabel 2.4 Simbol Proses

No	Simbol	Keterangan
1.		Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
2.		Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
3.		Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya/ tidak .
4.		Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5.		Menyatakan permulaan atau akhir suatu program.
6.		Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> .
7.		Menunjukkan bahwa data dalam <i>symbol</i> ini akan disimpan ke suatu <i>media</i> tertentu.
8.		Memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .

Input / Output symbol, menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*.

Tabel 2.5 Simbol *Input / Output*

No	Simbol	Keterangan
1.		Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
2.		Menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.

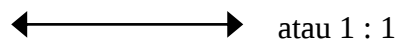
3.		Menyatakan <i>input</i> berasal dari pita magnetik satau <i>output</i> disimpan ke pita magnetik.
4.		Menyatakan <i>input</i> berasal dari disk atau <i>output</i> disimpan ke disk.
5.		Mencetak keluaran dalam bentuk dokumen (melalui printer).
6.		Mencetak keluaran dalam layar monitor.

2. Relasi

Relasi adalah hubungan antara suatu tabel dengan tabel lainnya dalam basis data relasi antara dua tabel dapat dikategorikan menjadi tiga macam, yaitu:

- a. Relasi satu ke satu.
 - b. Relasi satu ke banyak.
 - c. Relasi banyak ke banyak.
- a. Relasi Satu ke Satu (*One to One*)

Hubungan antara dua tabel adalah satu banding satu. Hubungan tersebut dapat ditunjukkan dengan tabel dan relasi antara keduanya dihubungkan dengan tanda panah tunggal.

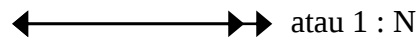


Gambar 2.1 *One to One Relation*

- b. Relasi Satu ke Banyak (*One to Many*)

Hubungan antara dua tabel adalah satu perbandingan banyak atau dapat pula dibalik dari banyak ke satu. Hubungan tersebut dapat

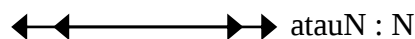
ditunjukkan dengan tabel dan relasi diantara keduanya dihubungkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.



Gambar 2.2 *One to Many Relation*

c. Relasi Banyak ke Banyak (*Many to Many*)

Hubungan antara dua tabel adalah banyak berbanding banyak. Hubungan tersebut dapat ditunjukkan dengan panah ganda dari masing – masing tabel.



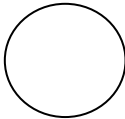
Gambar 2.3 *Many to Many Relation*

3. *Data Flow Diagram (DFD)*

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan asal data dan tujuan data yang keluar dari sistem, tempat penyimpanan data, proses apa yang menghasilkan data tersebut, serta interaksi antar data yang tersimpan dan proses yang dikenakan pada data tersebut.

Tabel 2.6 Simbol-simbol DFD

No	Simbol	Keterangan
1		Terminator : entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (<i>external entity</i>)

2		Data Store : Komponen ini digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda jamak
3		Proses : Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.
4		Alur Data : Komponen data flow (alur data) digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses

4. Diagram Konteks

Diagram konteks adalah suatu diagram alir yang menggambarkan seluruh jaringan, masukan dan keluaran. Sistem yang dimaksud adalah untuk menggambarkan sistem yang sedang berjalan. Mengidentifikasi awal dan akhir data, awal dan akhir yang masuk dan keluar sistem. Diagram ini merupakan gambaran umum sistem yang nantinya akan dibuat. Secara uraian dapat dikatakan bahwa diagram konteks itu berisi siapa saja yang memberikan data (*inputan*) ke sistem serta kepada siapa data informasi yang harus dihasilkan sistem.

5. Entity Relationship Diagram (ERD)

Entity Relationship Diagram dibuat untuk menunjukkan obyek-obyek (himpunan entitas) apa saja yang ingin dilibatkan dalam sebuah

basis data dan bagaimana hubungan terjadi diantara obyek-obyek tersebut. Dalam membentuk *entity relationship* ada dua komponen utama pembentuk model tersebut yaitu entitas (*entity*) dan relasi (*relation*). Entitas merupakan individu yang mewakili suatu yang nyata (eksistensinya) dan yang dibedakan dari suatu yang lain (Jogiyanto, 2010)

2.7 PHP

PHP adalah singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming* yaitu bahasa pemrograman yang diproses di sisi *server*. Fungsi utamanya *PHP* dalam membangun *website* adalah melakukan pengolahan data pada *database*. Data akan dimasukan ke *database*, di-edit, dihapus dan ditampilkan pada *website* yang diatur oleh *PHP* (Abdulloh, 2015)

2.8 MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun *aplikasi web* yang menggunakan *database* sebagai sumber dan pengolahan datanya (Ariefianto, 2013)

2.9 Visio 2007

Visio Merupakan program pengolahan desain grafik yang digunakan untuk mendesain model-model diagram sistem baik untuk perancangan jaringan, *database*, *software* aplikasi dan lain-lain. *Program aplikasi* ini berjalan dibawah *sistem operasi windows* (Chandra, 2013)

2.10 Basis Data

Basis data terdiri atas dua kata yaitu *basis* dan *data basis* diartikan sebagai gudang, tempat berkumpul, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu obyek yang direkam dalam bentuk huruf, *symbol*, teks, gambar, bunyi atau kombinasinya. Data ini juga diartikan sebagai bentuk yang masih mentah yang belum bercerita banyak, sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan sebuah informasi. Jadi basis data (*database*) adalah kumpulan *file-file* yang mempunyai kaitan antara yang satu dengan yang lainnya, sehingga membentuk suatu bangunan data untuk menginformasikan satu perusahaan atau *instansi* dalam batasan tertentu (Ariefanto, 2013).

2.11 Algoritma K-Nearest Neighbor

Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap obyek berdasarkan data pembelajaran yang jaraknya paling dekat dengan obyek tersebut. Prinsip kerja dari K-Nearest Neighbor (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (*neighbor*) terdekatnya dalam data pelatihan.

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah k buah yang paling dekat diambil. Titik

yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma k -nearest neighbor.

Ketepatan algoritma k -NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

Sesuai dengan prinsip kerja K-Nearest Neighbor yaitu mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga(neighbor) terdekatnya dalam data pelatihan. Persamaan dibawah ini menunjukkan rumus perhitungan untuk mencari jarak terdekat dengan d adalah jarak dan n adalah dimensi data (Agusta, 2014).

Kelebihan K-Nearest Neighbor:

1. Tahan terhadap data pelatihan yang memiliki noisy dan efektif apabila data latih nya besar.
2. Sangat sederhana dalam implementasi

3. Efektif untuk menghitung data dalam skala kecil
4. Beberapa parameter untuk acuan : jarak metrik dan k.

Kelemahan K-Nearest Neighbor:

1. K-NN perlu menentukan nilai dari parameter K (jumlah dari tetangga terdekat) sehingga untuk menyatakan jumlah tetangga terdekatnya lebih mudah.
2. Pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik
3. biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap data uji pada keseluruhan data latih.

Langkah-langkah algoritma *K-NN*:

1. Tentukan parameter K
Penentuan nilai parameter K bisa diubah-ubah sesuai dengan jumlah tetangga terdekat yang akan di klasifikasi.
2. Hitung jarak antara data uji dengan seluruh data latih.
Cara perhitungan akan menggunakan rumus *K-NN* di bawah yaitu dengan cara menghitung kedekatan antara data baru (data uji) dengan setiap data lama (data latih) yang sudah ada.
3. Hasi perhitungan jarak tersebut diurutkan secara *ascending*
4. Mengumpulkan kategori Y (klasifikasi *nearest neighbor* berdasarkan nilai K)
Kategori Y adalah kelas/objek

5. Dengan menggunakan kategori *nearest neighbor* yang paling mayoritas maka dapat di prediksi kategori objek/class

Berikut rumus *K-NN* di Persamaan (1)

$$d(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Keterangan:

x = Data Latih

y = Data Uji

d = Jarak

i = Variabel Data

n = Dimensi Data