**BAB II**

**TINJAUAN PUSTAKA**

1. **Tinjauan Studi**

Berikut ini adalah beberapa penelitian yang terkait dengan algoritma KNN, secara garis besar tinjauan studi untuk penelitian ini meliputi :

1. Penelitian yang dilakukan oleh Danang Aristianto, 2017, Penerapan Data Mining Untuk Mengklasifikasikan Penyakit Berdasarkan Wilayah Dengan Metode Klasifikasi Dan Algoritma c45. Puskesmas Purwakarta merupakan salah satu pusat pelayanan kesehatan masyarakat yang ada di daerah Purwakarta. Setiap harinya, Puskesmas Purwakarta melayani banyak pasien dari berbagai wilayah yang ada di daerah Purwakarta, jumlah pasien bisa mencapai ratusan orang untuk sekali pelayanan dalam satu hari dan menghasilkan data laporan harian kunjungan pasien yang sangat banyak. Setiap bulannya pihak Puskesmas Purwakarta melakukan sebuah penyuluhan penyakit kepada masyarakat di suatu wilayah. Namun penyuluhan yang dilakukan oleh pihak Puskesmas Purwakarta tersebut memiliki suatu masalah yaitu tidak tepat sasarannya penyakit yang disampaikan kepada masyarakat. Masalah tersebut terjadi karena penyuluhan yang dilakukan tidak berdasarkan data laporan harian kunjungan pasien yang ada. *Data mining* merupakan suatu teknik untuk menggali suatu informasi yang tersembunyi dari suatu kumpulan data. Metode yang digunakan untuk penelitian ini yaitu klasifikasi. Metode kalsifikasi ini bertujuan untuk mengklasifikasikan penyakit berdasarkan wilayah yang diperoleh dari data laporan harian kunjungan pasien. Algoritma yang digunakan adalah algoritma C4.5. Dengan metode dan algoritma tersebut nantinya akan menghasilkan suatu bentuk pohon keputusan yang bisa membantu pihak Puskesmas Purwakarta mengambil keputusan yang tepat sasaran dalam kegiatan penyuluhan penyakit di masyarakat. Pengujian yang dilakukan dengan membandingkan hasil pohon keputusan yang dihasilkan secara manual dan aplikasi. Berdasarkan hasil pengujian, aplikasi yang dibangun dapat membantu pihak Puskesmas Purwakarta untuk kegiatan penyuluhan penyakit kepada masyarakat menjadi tepat sasaran berdasarkan data laporan harian kunjungan pasien yang telah diolah menggunakan aplikasi.
2. Penelitian yang dilakukan oleh Dwi Intan Permatasari, 2017, Klasifikasi Status Ekonomi Keluarga Dengan Menggunakan Algoritma K-Nearest Neighbor Di Desa Pacewetan Kecamatan Pace Kabupaten Nganjuk. Penelitian ini dilatar belakangi oleh pendataan dan klasifikasi tingkat ekonomi warga Desa Pacewetan yang masih dilakukan secara manual, sehingga prosesnya kurang efisien. Berhubungan dengan pengontrolan kesejahteraan masyarakat pada Desa Pacewetan yang masih belum dilakukan secara mendetail, maka tingkat kesejahteraan warganya tidak terkontrol dengan baik. Penelitian ini menggunakan pendekatan penelitian kualitatif dikarenakan data kriteria diolah dengan menggunakan algoritma *K-Nearest Neighbor* untuk mendapatkan hasil klasifikasi status ekonomi keluarga di Desa Pacewetan. Kriteria yang digunakan dalam penelitian adalah data pekerjaan, penghasilan perbulan, pendidikan terakhir, jumlah anggota rumah tangga, jumlah anak bersekolah, luas bangunan, jenis lantai, jenis dinding, daya listrik,jenis bahan bakar memasak, kepemilikan kendaraan, dan kepemilikan asset. Kesimpulan hasil penelitian ini adalah : (1) Menghasilkan aplikasi klasifikasi status ekonomi keluarga di Desa Pacewetan. (2) Dengan penerapan algoritma *K-Nearest Neighbor* dapat mengklasifikasikan status ekonomi keluarga. Berdasarkan hasil pengujian dengan *5-fold cross* *validation* didapatkan rata-rata presentase akurasi sebesar 98.8%, sehingga dapat disimpulkan bahwa algoritma *K-Nearest Neighbor* memiliki kinerja yang baik.
3. Penelitian yang dilakukan oleh Yampi R Kaesmetan dan Jusrianto A Johannis, 2017, Klasifikasi Status Gizi Balita Di Kelurahan Oesepa Barat Menggunakan Metode K-Nearest Neighbor. Status gizi adalah keadaan tubuh karena pola konsumsi makanan dan penggunaan nutrisi yang disesuaikan dengan kebutuhan tubuh. Penentuan status gizi anak berguna untuk mengetahui kondisi BB berdasarkan gizi BB / U (Bobot Menurut Usia), TB / U (Tinggi menurut Umur), BB / TB (Bobot dengan Tinggi). Sistem yang dirancang untuk penentuan status gizi pada anak menggunakan K-NN (K-Nearest Neighbor), dimana metode K-NN (K-Nearest Neighbor) merupakan metode untuk mengklasifikasikan data atau mengklasifikasikan data uji yaitu kelas latih ke beberapa Tetangga terdekat dengan menggunakan rumus perhitungan jarak.. Variabel yang digunakan dalam sistem ini didasarkan pada data antropometrik atau pengukuran tubuh manusia, yaitu U (Age), BB (Bobot), TB (Tinggi), LK (Head Circumference). Perhitungan manual menggunakan 10 data sampel kemudian dibuat rangking dan diuji dengan Aplikasi Matlab dan hasilnya sama dengan perhitungan manual. Hasil yang telah didapatkan diaplikasikan pada pemetaan menggunakan Arcview dan JOSM. Hasil pemetaan memberikan informasi lokasi rawan malnutrisi di kawasan Desa Oesapa Barat. metode K-*Nearest Neighbor* dalam pengklasifikasian status gizi dengan menggunakan formulasi perhitungan jarak eucledian memiliki kinerja yang baik. Hal ini dapat dilihatdari hasil penguji kinerja system yang dihasilkan oleh jarak Eucledian dengan nilai akurasi sebesar 100%.
4. **Tinjauan Teori**

**2.2.1 Penyakit**

Penyakit adalahsuatu keadaan tidak normal dari tubuh atau pikiran yang menyebabkan ketidak nyamanan, disfungsi, atau kesukaran terhadap orang yang dipengaruhinya. Penyakit dibedakan menjadi penyakit menular, penyakit tidak menular, dan penyakit kronis.

Menurut Wahyudin Rajab (1872), Penyakit/sakit adalah kegagalan mekanisme adaptasi suatu organisme untuk bereaksi secara tepat terhadap rangsangan atau tekanan sehingga timbul gangguan pada fungsi atau struktur organ atau sistem tubuh. Sedangkan menurut Daldiyono Hardjodisastro (2006), penyakit adalah sesuatu yang abnormal (misalnya rasa sakit) yang merugikan yang terdapat pada seseorang yang semula sehat. Definisi lain yang dimuat dalam Butterworth Medical Dictionary, agaknya baik untuk dikutip disini agar pengertian tentang istilah penyakit menjadi jelas.

Atribut yang digunakan untuk mengklasifikasikan penyakit adalah sebagai berikut :

**Tabel 2.1.** Atribut klasifikasi penyakit

| **No** | **Atribut** | **Nilai Atribut** |
| --- | --- | --- |
| 1 | Jenis kelamin | - Laki-Laki  - Perempuan |
| 2 | Umur | - < 5 Tahun  - 6 – 14 Tahun  - 15 – 20 Tahun  - 20 – 40 Tahun  - > 40 Tahun |
| 3 | Status | - Lama  - Baru |
| 4 | Pekerjaan | - IRT  - Petani  - Buruh  - Pegawai Swasta  - PNS  - Guru  - Karyawan  - Honorer  - Wiraswasta  - Pedagang  - Supir |
| 5 | Wilayah | - Kelurahan Pauwo  - Kelurahan Tumbihe  - Kelurahan Oluhuta  - Kelurahan Oluhuta Utara  - Kelurahan Padengo |
| 6 | Nama penyakit | - Common Cold/Nasofaringitis Akut  - Influenza  - Essential (primari) Hipertension  - Diare  - Dermatitis Nummular  - Dyspepsia  - Headache  - TBC Paru dengan BTA+  - Dermatitis Kontak Alergika |
| 7 | Kategori | - Ringan  - Sedang  - Berat |

Sumber: Puskesmas Kabila, 2016

**2.2.2 Data Mining**

Data Mining (Witten, 2011) didefinisikan sebagai proses penemuan pola dalam data. Menurut Daryl Pregibons dalam (Gorunescu, 2011) “Data mining adalah perpaduan dari ilmu statistik, kecerdasan buatan, dan penelitian bidang database”. Nama data mining berasal dari kemiripan antara pencarian informasi yang bernilai dari database yang besar dengan menambang sebuah gunung untuk sesuatu yang bernilai (Sumathi, 2006). Keduanya memerlukan penyaringan melalui sejumlah besar material, atau menyelidiki dengan cerdas untuk mencari keberadaan sesuatu yang disebut bernilai tadi. Istilah lain dari data (Han, 2006) yaitu *knowledge mining from databases*, *knowledge extraction*, data/pattern analysis, *data archeology*, dan data *dredging*. Banyak yang menggunakan data mining sebagai istilah popular dari KDD.

Data mining (Maimon, 2005) merupakan inti dari proses *Knowledge Discovery in Database* (KDD). KDD adalah proses terorganisir untuk mengidentifikasi pola yang valid, baru, berguna, dan dapat dimengerti dari sebuah data set yang besar dan kompleks.

**2.2.3 Proses Tahapan Data Mining**

Istilah data mining dan knowledge discovery databases (KDD) sering digunakan secara bergantian untuk menjelaskan proses penggalian informasi dalam database. Secara garis besar prosess KDD dapat dijelaskan sebagai berikut (Maimon, 2005):

1. Pembentukan pemahaman domain aplikasi

Pada tahap ini menentukan tujuan dari end-user dan bagian terkait dimana KDD dilakukan.

2. Memilih dan membuat data set dimana proses penemuan *knowledge* akan dilakukan.

Penentuan data yang akan digunakan untuk proses KDD dilakukan pada tahap ini. Mencari data yang tersedia, memperoleh data tambahan yang dibutuhkan, mengintegrasikan semua data untuk KDD ke dalam sebuah data set, termasuk atribut yang diperlukan dalam proses KDD.

3. *Preprocessing* dan *cleansing*

Dalam tahap ini kehandalan data ditingkatkan. Termasuk data cleansing, seperti menangani data yang tidak lengkap, menghilangkan gangguan atau outlier.

4. Transformasi data

Pada tahap ini, membuat data menjadi lebih baik menggunakan metode reduksi dimensi dan transformasi atribut.

5. Memilih tugas data mining yang cocok

Pada tahap ini ditentukan tipe data mining yang akan digunakan, apakah klasifikasi, regresi, atau clustering, tergantung pada tujuan KDD dan tahap sebelumnya.

6. Memilih algoritma data mining

Pada tahap ini dilakukan pemilihan algoritma yang paling tepat untuk menemukan pola.

7. Penggunaan algoritma data mining

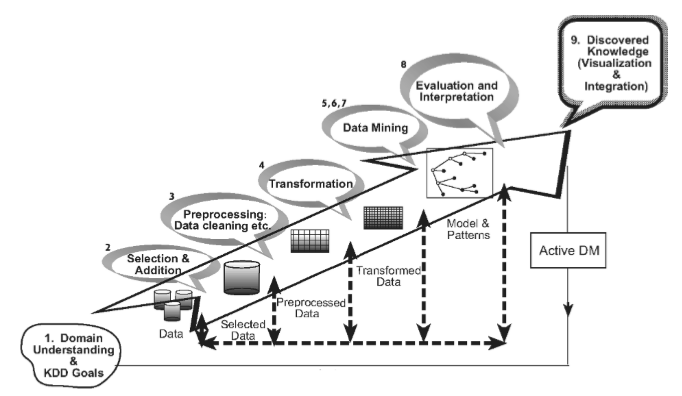
Pada tahap ini dilakukan implementasi dari algoritma data mining yang telah ditentukan pada tahap sebelumnya.

8. Evaluasi

Pada tahap ini dilakukan evaluasi dan penerjemahan dari pola yang didapat.

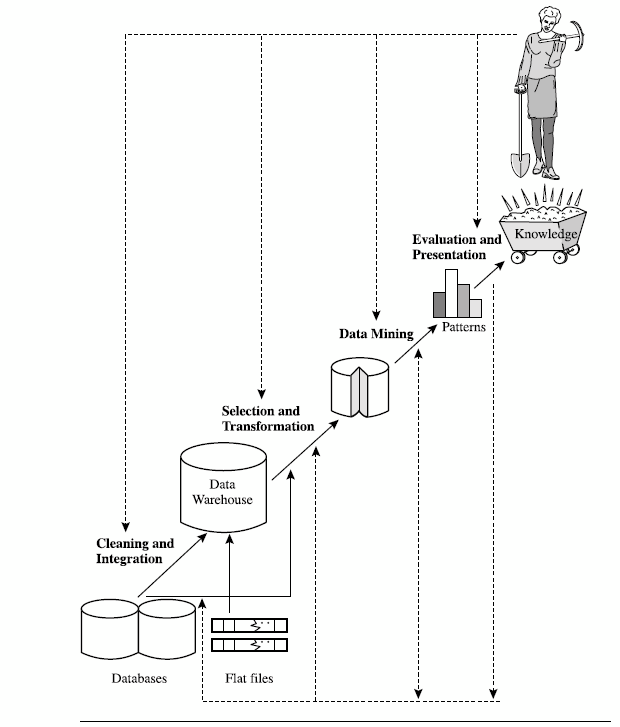
9. Penggunaan pengetahuan yang didapat

Pada tahap ini, pengetahuan dimasukkan ke sistem lain dan mengaktifkan sistem tersebut serta mengukur hasilnya. (Maimon, 2005)



**Gambar 2.1.** Proses KDD

Sedangkan menurut (Han & Kamber, 2006) tahapan dalam KDD adalah sebagai berikut:



**Gambar 2.2.**  Tahap-Tahap Data Mining

1. Pembersihan data (data *cleaning*) Pembersihan data merupakan proses menghilangkan noise dan data yang tidak konsisten atau data tidak relevan. Pada umumnya data yang diperoleh, baik dari database suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesa data mining yang dimiliki. Data-data yang tidak relevan itu juga lebih baik dibuang. Pembersihan data juga akan mempengaruhi performasi dari teknik data mining karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

2. Integrasi data (data *integration*) Integrasi data merupakan penggabungan data dari berbagai database ke dalam satu database baru. Tidak jarang data yang diperlukan untuk data mining tidak hanya berasal dari satu database tetapi juga berasal dari beberapa database atau file teks. Integrasi data dilakukan pada atribut-aribut yang mengidentifikasikan entitas-entitas yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

3. Seleksi Data (Data *Selection*) Data yang ada pada database sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari database. Sebagai contoh, sebuah kasus yang meneliti faktor kecenderungan orang membeli dalam kasus market basket analysis, tidak perlu mengambil nama pelanggan, cukup dengan id pelanggan saja.

4. Transformasi data (Data *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. Sebagai contoh beberapa metode standar seperti analisis asosiasi dan clustering hanya bisa menerima input data kategorikal. Karenanya data berupa angka numerik yang berlanjut perlu dibagibagi menjadi beberapa interval. Proses ini sering disebut transformasi data.

5. Proses mining, Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

6. Evaluasi pola (*pattern evaluation*), Untuk mengidentifikasi pola-pola menarik kedalam *knowledge based* yang ditemukan. Dalam tahap ini hasil dari teknik data mining berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai hipotesa ada beberapa alternatif yang dapat diambil seperti menjadikannya umpan balik untuk memperbaiki proses data mining, mencoba metode data mining lain yang lebih sesuai, atau menerima hasil ini sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

7. Presentasi pengetahuan (*knowledge presentation*), Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Tahap terakhir dari proses data mining adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisis yang didapat. Ada kalanya hal ini harus melibatkan orang-orang yang tidak memahami data mining. Karenanya presentasi hasil data mining dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses data mining. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan hasil data mining (Han, 2006).

**2.2.4 Kelompok Data Mining**

Berdasarkan tugasnya, data mining dikelompokkan menjadi (Larose, 2005):

1. Deskripsi

Mencari cara untuk menggambarkan pola dan trend yang terdapat dalam data. Sebagai contoh, seorang pengumpul suara mengungkap bukti bahwa mereka yang diberhentikan dari jabatannya saat ini, akan kurang mendukung dalam pemilihan presiden. Untuk deskripsi ini bisa dilakukan dengan *exploratory* data *analysis*, yaitu metode grafik untuk menelusuri data dalam mencari pola dan tren. (Larose, 2005)

2. Estimasi

Estimasi mirip seperti klasifikasi tapi variable sasaran adalah numerik. Model dibuat menggunakan record yang lengkap, juga ada variable targetnya. Kemudian untuk data baru, estimasi nilai variable target dibuat berdasarkan nilai prediktor. Contoh, untuk estimasi tekanan darah pada pasien, variable prediktornya umur, jenis kelamin, berat badan, dan tingkat sodium darah. Hubungan antara tekanan darah, dan variable prediktor pada data training akan menghasilkan model kemudian diaplikasikan pada data baru. Untuk melakukan estimasi bisa digunakan *neural network* atau metode *statistic* seperti *point estimation* dan *conﬁdence interval estimations*, simple linear *regression* dan *correlation*, dan *multiple regression* (Larose, 2005).

3. Prediksi

Prediksi mirip seperti klasifikasi dan estimasi, tapi hasilnya untuk memprediksi masa depan. Contoh, memprediksi harga barang tiga bulan mendatang, memprediksi presentasi kenaikan angka kematian karena kecelakaan tahun mendatang jika kecepatan berkendara dinaikkan. Metode dan teknik untuk klasifikasi dan estimasi, jika cocok, bisa juga digunakan untuk prediksi, termasuk metode statistik. Algoritma untuk prediksi antara lain *regression tree* dan model *tree* (Han, 2006).

4. Klasifikasi

Dalam klasifikasi, sasarannya adalah variable kategori, misalkan atribut penghasilan, yang bisa dikategorikan menjadi tiga kelas atau kategori yaitu, tinggi, sedang, dan rendah. Model data mining membaca sejumlah besar record tiap record berisi informasi pada variable target. Contoh, dari sebuah data set misalkan mau mengklasifikasikan penghasilan seseorang yang datanya tidak terdapat pada dataset, berdasarkan karakteristik yang berhubungan dengan orang itu seperti, umur, jenis kelamin, dan pekerjaan. Tugas klasifikasi ini cocok untuk metode dan teknik data mining. Algoritma akan mengolah dengan cara membaca data set yang berisi variable predictor dan variable taget yang telah diklasifikasikan, yaitu penghasilan. Di sini algoritma (*software*) “mempelajari” kombinasi variable mana yang berhubungan dengan penghasilan yang mana. Data ini disebut *training set*. Kemudian algoritma akan melihat ke data baru yang belum termasuk klasifikasi manapun. Berdasarkan klasifikasi pada data set, kemudian algoritma akan memasukkan data baru tersebut ke dalam klasifikasi yang mana. Misalkan seorang professor wanita berusia 63 tahun bisa jadi diklasifikasikan ke dalam kelas penghasilan tinggi. Algoritma klasifikasi yang banyak digunakan secara luas untuk klasifikasi antara lain, *decision tree*, *bayesian classifier*, dan *neural network* (Gorunescu, 2011)

5. *Clustering*

*Clustering* mengacu pada pengelompokkan record-record, observasi, atau kasus-kasus ke dalam kelas-kelas dari objek yang mirip. Pada *clustering* tidak ada variabel sasaran. Sebuah cluster adalah koleksi record yang mirip satu sama lain, dan tidak mirip dengan *record* pada *cluster*. Tidak seperti klasifikasi, pada *clustering* tidak ada variable target. *Clustering* tidak menglasifikasi atau mengestimasi atau memprediksi tetapi mencari untuk mensegmentasi seluruh data set ke subgroup yang *relative* sejenis atau *cluster*, dimana kemiripan record di dalam cluster dimaksimalkan dan kemiripan dengan record di luar cluster diminimalkan. Contoh *clustering*, untuk akunting dengan tujuan audit untuk mensegmentasi financial behaviour ke dalam kategori ramah dan curiga, sebagai alat reduksi dimensi ketika data set memiliki ratusan atribut, untuk clustering ekspresi gen, dimana kuantitas gen bisa terlihat mempunyai behavior yang mirip. Algoritma untuk *clustering* antara lain, *hierarchical agglomerative clustering, Bayesian clustering, self – organizing feature maps, growing hierarchical self - organizing maps* (Wu, 2009).

6. Asosiasi

Tugas asosiasi untuk data mining adalah kegiatan untuk mencari atribut yang “*go together*.” Dalam dunia bisnis, asosiasi dikenal sebagai affinity analysis atau market basket analysis, tugas asosiasi adalah membuka rules untuk pengukuran hubungan antara dua atribut atau lebih. Contoh asosiasi, prediksi degradasi dalam jaringan komunikasi, menemukan barang apa di supermarket yang dibeli bersama dengan barang lain yang tidak pernah dibeli bersama, menemukan proporsi kasus dimana obat baru akan memperlihatkan efek samping yang berbahaya. Untuk menemukan association rules, bisa dilakukan dengan algoritma a priori dan algoritma GRI (*Generalized Rule Induction*) (Larose, 2005) .

**2.2.5 Algoritma Klasifikasi Data Mining**

Klasifikasi (Han, 2006) adalah proses penemuan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui.

Klasifikasi data terdiri dari 2 langkah proses. Pertama adalah learning (fase training), dimana algoritma klasifikasi dibuat untuk menganalisa data training lalu direpresentasikan dalam bentuk rule klasifikasi. Proses kedua adalah klasifikasi, dimana data tes digunakan untuk memperkirakan akurasi dari rule klasifikasi. (Han, 2006).

Proses klasifikasi didasarkan pada empat komponen: (Gorunescu, 2011)

a. Kelas

Variabel dependen yang berupa kategorikal yang merepresentasikan “label” yang terdapat pada objek. Contohnya: resiko penyakit jantung, resiko kredit, *customer loyalty*, jenis gempa.

b. Predictor

Variabel independen yang direpresentasikan oleh karakteristik (atribut) data. Contohnya: merokok, minum alkohol, tekanan darah, tabungan, aset, gaji.

c. Training dataset

Satu set data yang berisi nilai dari kedua komponen di atas yang digunakan untuk menentukan kelas yang cocok berdasarkan predictor.

d. Testing datatest

Berisi data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi

Berikut ini adalah algoritma klasifikasi yang banyak digunakan secara luas: (Gorunescu, 2011)

1. *Decision/classiﬁcation trees*

Decision tree digunakan untuk memprediksi keanggotaan suatu objek ke dalam kategori (kelas) yang berbeda, berdasarkan variabel prediktor. Algoritma decision tree yang dikenal luas antara lain Hunt, CART (C&RT), ID3, C4.5 & C5.0, SLIQ, SPRINT, QUEST, DTREG, THAID, CHAID (Gorunescu, 2011).

2. *Bayesian classiﬁers/ Naïve Bayes classiﬁers*

Klasifikasi Bayes (Kusrini, 2009) adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas. Klasifikasi Bayes juga dikenal dengan Naïve Bayes, idiot’s Bayes, simple Bayes, dan independence Bayes (Wu, 2009). Klasifikasi Bayes didasarkan pada teorema Bayes, diambil dari nama seorang ahli matematika yang juga menteri Prebysterian Inggris, Thomas Bayes (1702-1761).

3. *Neural networks*

Neural network adalah (Han, 2006) satu set unit input/output yang terhubung dimana tiap relasinya memiliki bobot. Neural Network dimaksudkan untuk mensimulasikan perilaku sistem biologi susunan syaraf manusia (Alpaydin, 2010).

4. Analisa Statistik

Analisa statistik selama ini juga melakukan hal yang sama seperti data mining. Beberapa metode estimasi dan prediksi tradisional merupakan turunan dari analisa statistik, antara lain linear regression dan multiple regression (Larose, 2005).

5. Algoritma Genetika

Algoritma genetika merepresentasikan teknik identifikasi solusi perkiraan untuk masalah optimasi dan pencarian. Algoritma genetika merupakan tipe algoritma evolusioner yang paling populer, berbasis pada populasi solusi potensial mengguinakan mekanisme spesifik yang inspirasi dari evolusi biologi (genetika alamiah), seperti kromosom, reproduksi, rekombinasi, seleksi*, survival of fittest* (Gorunescu, 2011).

6. *Rough sets*

Rough Set dibuat oleh Zdzislaw Pawlak pada awal tahun 1980, dalam rangka mengungkap secara matematis konsep kesamaran, tujuan utamanya adalah menjadi proses otomatisasi transformasi data ke dalam pengetahuan. Rough set merupakan pendekatan matematis untuk pengetahuan yang tidak sempurna, hal ini penting dalam logika fuzzy. Konsep rough set secara umum bisa didefinisikan dengan 2 topologi, yaitu *interior* dan *closure*. Ide dasar *rough set* terdapat pada keyataan bahwa, berdasarkan satu set objek, satu set atribut dan nilai keputusan, seseorang dapat membuat rule untuk menemukan perkiraan *upper* dan *lower*, dan daerah batasan (*boundary region*) dari set objek. Setelah rule dibuat, maka objek baru bisa dengan mudah diklasifikasikan ke dalam salah satu daerah (region) (Gorunescu, 2011).

7. K-*nearest neighbor*

K-*nearest neighbor* (kNN) adalah klasifikasi yang menyimpan semua data training dan melakukan klasifikasi dengan cara membandingkan antara atribut data baru yang paling cocok dengan atribut record yang terdapat pada data training. KNN adalah kasus khusus dalam instance-based learning. Ini termasuk case-based reasoning, yang menangani data simbol. KNN juga merupakan contoh teknik lazy learning, yaitu teknik yang menunggu sampai pertanyaan (*query*) datang agar sama dengan data training (Wu, 2009).

8. Metode *Rule Based*

Klasifikasi *rule based* artinya proses penggunaan data training dengan objek berlabel dimana rule akan diekstrak untuk menghasilkan klasifikasi Gorunescu, 2011).

9. Memory *based reasoning*

10. *Support vector machines* (SVM)

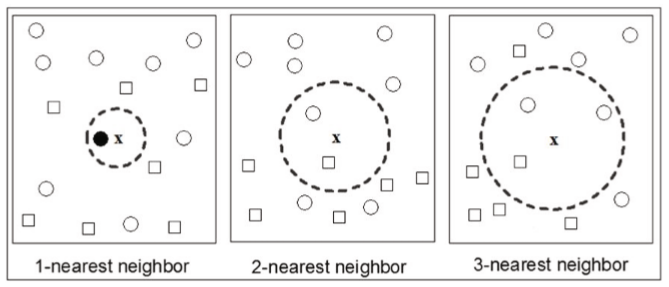
*Support vector machines* (SVM) dibuat oleh Vapnik untuk mengimplementasikan secara konstruktif prinsip dari teori statistical learning. Dalam kerangka statistical learning, learning artinya mengestimasi sebuah fungsi dari data training. Untuk melakukan ini, sebuah machine learning harus memilih satu fungsi untuk meminimalkan sejumlah resiko yang fungsi estimasinya berbeda dengan fungsi kenyataan (yang belum diketahui). Resiko tergantung pada kompleksitas fungsi-fungsi yang dipilih dari training set. Kemudian learning machine harus menemukan fungsi terbaik, ditentukan oleh kompleksitasnya. SVM cukup populer untuk penggunaan klasifikasi karena kelebihannya antara lain dari segi cara kerja, SVM baik untuk klasifikasi, tidak tergantung pada jumlah fitur dan bisa mengatasi masalah dimensi. Dari segi komputasi, SVM dapat melakukan proses training dengan cepat dan ini berguna dalam teknik learning ketika mengadapi masalah ketidaktegasan (Maimon, 2005).

**2.2.6 Algoritma K-*Nearest Neighbor***

K-Nearest Neighbor (kNN) termasuk kelompok instance-based learning. Algoritma ini juga merupakan salah satu teknik lazy learning. KNN dilakukan dengan mencari kelompok k objek dalam data training yang paling dekat (mirip) dengan objek pada data baru atau data testing (Wu, 2009).

Algortima KNN merupakan sebuah metode untuk melakukan klasifikasi terhadap obyek baru berdasarkan (K) tetangga terdekatnya. (Gorunescu, 2011).

Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru bedasarkan atribut dan training sample. Classifier tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik query,akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik query. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek. Algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari query instance yang baru.



**Gambar 2.3** KNN dengan Nilai K tetangga (k=1,2,3)

Algoritma metode KNN sangatlah sederhana, bekerja berdasarkan jarak terpendek dari query instance ke training sample untuk menentukan KNN-nya. Training sample diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi training sample. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut.

Pada fase training, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data (yang klasifikasinya tidak diketahui).  Jarak dari vektor baru yang ini terhadap seluruh vektor training sample 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 semakin kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus dimana klasifikasi diprediksikan berdasarkan training datayang paling dekat (dengan kata lain, k = 1) disebut algoritma nearest neighbor.

KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap training data yang memiliki banyak noise dan efektif apabila training data-nya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat), training berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap  query instance pada keseluruhan  training sample.

Langkah-langkah untuk menghitung metode Algoritma *K*-*Nearest Neighbor*:

a. Menentukan Parameter K (Jumlah tetangga paling dekat).

b. Menghitung kuadrat jarak Euclid (*queri instance*) masing-masing objek terhadap data sampel yang diberikan.

c. Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak Euclid terkecil.

d. Mengumpulkan kategori Y (*Klasifikasi Nearest Neighbor*)

e. Dengan menggunakan kategori *Nearest Neighbor* yang paling mayoritas maka dapat diprediksi nilai queri instance yang telah dihitung.

Ada banyak cara untuk mengukur jarak kedekatan antara data baru dengan data lama (data training), diantaranya euclidean distance dan manhattan distance (city block distance), yang paling sering digunakan adalah euclidean distance (Bramer, 2007), yaitu:

.................... (2.1)

Dimana :

a = a1,a2, …, an, dan;

b = b1, b2, …, bn mewakili n nilai atribut dari dua record.

Untuk atribut dengan nilai kategori, pengukuran dengan euclidean distance tidak cocok. Sebagai penggantinya, digunakan fungsi sebagai berikut (Larose, 2006):

*Different (ai,bi)* { 0 jika ai = bi dan 1 jika ai # bi .............. (2.2)

dimana ai dan bi adalah nilai kategori.

Jika nilai variabel antara dua record yang dibandingkan sama maka nilai jaraknya 0, jika nilai variabel mempunya kemiripan atau kedekatan maka diberi bobot 0,5, sebaliknya jika nilai variabel berbeda maka akan diberi bobot 1 artinya tidak mirip sama sekali. Misalkan variabel status dengan nilai menikah dan menikah, maka nilai kedekatannya 0, jika menikah dan janda maka nilai kedekatannya antara 0,5, sedangkan jika nilai menikah dan belum menikah maka nilai kedekatannya adalah 1.

Untuk menghitung kemiripan kasus, digunakan rumus (Kusrini, 2009): ..................... (2.3)

Keterangan :

T = Kasus baru

S = Kasus yang ada dalam penyimpanan

n = Jumlah atribut dalam tiap kasus

Ti = Kasus baru dalam Atribut individu antara 1 sampai dengan n

Si = Kasus yang ada dalam penyimpanan Atribut individu antara 1 sampai dengan n

f = Fungsi similarity atribut i antara kasus T dan kasus S

wi = Bobot yang diberikan pada atribut ke-i kedalam Atribut individu antara 1 sampai dengan n

**2.2.7 Metode Evaluasi *Confusion Matrix***

*Confusion Matrix* adalah tool yang digunakan untuk evaluasi model klasifikasi untuk memperkirakan obejek yang benar atau salah (Gorunescu, 2011). Sebuah Matrix dari prediksi yang akan dibandingkan dengan kelas yang asli dari inputan atau dengan kata lain berisi nilai actual dan prediksi pada klasifikasi.

**Tabel 2.2** *Confusion Matrix*

|  |  |  |
| --- | --- | --- |
| ***Classification*** | *Predicted Class* | |
| *Class = Yes* | *Class = No* |
| *Class = Yes* | a *(true positive-*TP*)* | b *(false negative-*FN*)* |
| *Class = No* | c (*false positive-*FP*)* | d *(true negative*-TN*)* |

Keterangan Tabel:

1. *True Positives* merupakan jumlah *record* positif yang diklasifikasikan sebagai positif.
2. *False Positives* merupakan jumlah *record* negatif yang diklasifikasikan sebagai Positif.
3. *False Negatives* merupakan jumlah *record* positif yang di klasifikasikan sebagai negative.
4. *True Negatives* merupakan jumlah *record* negatif yang diklasifikasikan sebagai negative.

Rumus untuk menghitung tingkat akurasi pada matrix adalah:

**2.2.8 Penerapan Algoritma K-*Nearest Neighbor***

Berikut contoh penerapan algoritma K-NN dengan kasus penentuan resiko kredit kepemilikan kendaraan bermotor (Leidiyana, 2013). Tahapan sebagai berikut :

1. Penentuan atribut

Atribut dan nilai atribut yang digunakan dapat dilihat pada tabel berikut :

**Tabel 2.3** Penentuan Atribut dan Nilai Atribut

|  |  |  |
| --- | --- | --- |
| **No** | **Atribut** | **Nilai Atribut** |
| 1 | Status Perkawinan | Menikah  Belum menikah  Janda/duda |
| 2 | Jumlah Tanggungan | Tidak ada  1 orang  2 orang  > 2 orang |
| 3 | Lama tinggal | > 5 Tahun  3 – 5 Tahun  1 – 3 Tahun  < 1 Tahun |
| 4 | Masa Kerja/usaha | > 5 Tahun  2 – 5 Tahun  < 2 Tahun |
| 5 | Penghasilan perbulan | > 3 x angsuran  > 2 x angsuran  > 1 x angsuran  < 1 x angsuran |
| 6 | Pembayaran pertama | > 30%  20 -30%  10 -20%  < 10% |

1. Penentuan Bobot Atribut

Untuk mengukur jarak antar atribut, akan diberikan bobot pada atribut. Bobot jarak ini diberikan nilai antara 0 sampai dengan 1. Nilai 0 artinya jika atribut tidak berpengaruh dan sebaliknya nilai 1 jika atribut sangat berpengaruh.

**Tabel 2.4.** Pembobotan Atribut

|  |  |  |
| --- | --- | --- |
| **No.** | **Atribut** | **Bobot** |
| 1 | Status Perkawinan | 0,5 |
| 2 | Jumlah Tanggungan | 1 |
| 3 | Lama tinggal | 1 |
| 4 | Masa Kerja/usaha | 1 |
| 5 | Penghasilan perbulan | 1 |
| 6 | Pembayaran pertama | 1 |

1. Penentuan Bobot Nilai Kedekatan Atribut

**Tabel 2.5**. Bobot Kedekatan Nilai Atribut Status Perkawinan

|  |  |  |  |
| --- | --- | --- | --- |
|  | Menikah | Belum Menikah | Janda/Duda |
| Menikah | 0 | 1 | 0,5 |
| Belum Menikah | 1 | 0 | 0,5 |
| Janda/Duda | 0,5 | 0,5 | 0 |

**Tabel 2.6** Bobot Kedekatan Nilai Atribut Jumlah Tanggungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Tidak Ada | 1 Orang | 2 Orang | >2 orang |
| Tidak Ada | 0 | 0,5 | 0,5 | 1 |
| 1 Orang | 0,5 | 0 | 0,5 | 1 |
| 2 Orang | 0,5 | 0,5 | 0 | 0,5 |
| >2 orang | 1 | 1 | 0,5 | 0 |

Pembobotan kedekatan nilai atribut dilakukan untuk 6 atribut prediktor dengan cara seperti di atas. Setelah itu hitung kemiripannya.

1. Menentukan data training dan data testing

Misal sebuah data konsumen baru akan diklasifikasi apakah bermasalah atau tidak dalam penerima raskin maka dilakukan perhitungan kedekatan antara kasus baru dibandingkan dengan data kasus lama (data training). Kasus nomor 1 dan nomor 2 sebagai sampel data training dan kasus baru sebagai data testing dengan nilai atribut seperti pada tabel berikut :

**Tabel 2.7** Sampel Data Set

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Kasus | Status Perkawinan | Jumlah Tanggungan | Lama tinggal | Masa Kerja/usaha | Penghasilan perbulan | Pembayaran pertama | Remark |
| 1 | Menikah | tidak ada | 3-5 | <2 | >2x angsuran | 10-20% | Bad |
| 2 | Menikah | >2 | 3-5 | >5 | >3x angsuran | 10-20% | Good |
| Baru | blm menikah | tidak ada | >5 | <2 | >2x angsuran | 10-20% | ? |

1. Menghitung kedekatan kasus antara data training dan data testing

Untuk menghitung kedekatan kasus antara data training dan data testing di atas, digunakan persamaan (2.3). Untuk mengetahui apakah data testing termasuk kalasifikasi bad atau good, dapat dilakukan langkah-langkah sebagai berikut :

1. Menghitung kedekatan antara data baru dengan data nomor 1.

**Tabel 2.8**. Kedekatan Kasus Baru dengan Kasus Nomor 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Kasus 1** | **Kasus Baru** | **Kedekatan** | **Bobot** |
| Status Perkawinan | Menikah | Blm menikah | 1(a) | 0,5(b) |
| Jumlah Tanggungan | Tidak ada | Tidak ada | 0(c) | 1(d) |
| Lama tinggal | 3-5 | >5 | 0,5(e) | 1(f) |
| Masa Kerja/usaha | <2 | <2 | 0(g) | 1(h) |
| Penghasilan perbulan | >2x angsuran | >2x angsuran | 0(i) | 1(j) |
| Pembayaran pertama | 10-20% | 10-20% | 0(k) | 1(l) |

2. Menghitung kedekatan antara data baru dengan data nomor 2.

**Tabel 2.9**. Kedekatan Kasus Baru dengan Kasus Nomor 2

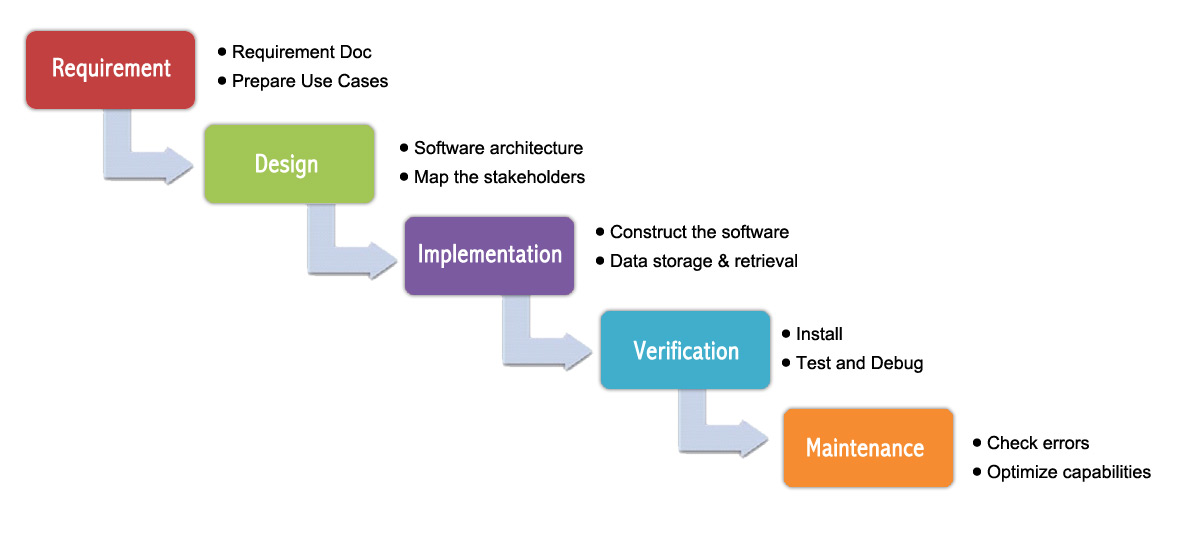
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atribut** | **Kasus 2** | **Kasus Baru** | **Kedekatan** | **Bobot** |
| Status Perkawinan | Menikah | Blm menikah | 1(a) | 0,5(b) |
| Jumlah Tanggungan | >2 | Tidak ada | 1(c) | 1(d) |
| Lama tinggal | 3-5 | >5 | 0,5(e) | 1(f) |
| Masa Kerja/usaha | >5 | <2 | 1(g) | 1(h) |
| Penghasilan perbulan | >3x angsuran | >2x angsuran | 0,5(i) | 1(j) |
| Pembayaran pertama | 10-20% | 10-20% | 0(k) | 1(l) |

3. Memilih kasus dengan kedekatan terdekat. Dari langkah 1 dan 2 dapat diketahui bahwa nilai terendah adalah kasus nomor 1. Berarti kasus yang terdekat dengan kasus baru adalah kasus nomor 1 pada data training.

4. Menggunakan klasifikasi dari kasus dengan kedekatan terdekat. Berdasarkan hasil yang diperoleh pada langkah 3, maka klasifikasi dari 2 kasus yang akan digunakan untuk mengklasifikasi kasus baru, yaitu kemungkinan konsumen baru tersebut akan **bermasalah (*bad*)** dalam pembayaran angsurannya.

**2.2.9 Siklus Hidup Pengembangan Sistem**

Menurut Jogiyanto (2005:41), Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara. Bila operasi yang sudah dikembangkan masih timbul kembali permasalahan-permasalahan yang tidak dapat diatasi dalam tahap pemeliharaan, maka perlu dikembangkan kembali suatu sistem untuk mengatasinya dan proses ini kembali ke tahap yang pertama, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem *(systems life cycle).* Daur atau siklus hidup dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah didalam tahapan tersebut dalam proses pengembangannya. Berikut langkah-langkah yang digunakan :



**Gambar 2.4** Siklus Hidup Pengembangan Sistem

**2.2.9.1 Perencanaan Sistem**

Kebijakan untuk mengembangkan sistem informasi dilakukan oleh manajemen puncak karena menginginkan untuk meraih kesempatan-kesempatan yang ada yang tidak dapat diraih oleh sistem lama atau sistem yang lama mempunyai banyak kelemahan-kelemahan yang perlu diperbaiki. Setelah manajemen puncak menetapkan kebijakan untuk mengembangkan sistem informasi, sebelum sistem ini sendiri dikembangkan, maka perlu direncanakan terlebih dahulu dengan cermat. Perencanaan sistem ini menyangkut estimasi dari kebutuhan-kebutuhan fisik, tenaga kerja, dan dana yang dibutuhkan untuk mendukung pengembangan sistem ini serta untuk mendukung operasinya setelah diterapkan.

Selama fase perencanaan sistem, hal yang perlu dipertimbangkan adalah :

1. Faktor-Faktor Kelayakan (*Feasibility Factors*)yang berkaitan dengan kemungkinan berhasilnya sistem informasi yang dikembangkan dan digunakan.
2. Faktor-Faktor Strategis (*Strategic Factors*)yang berkaitan dengan pendukung sistem informasi dari sasaran bisnis dipertimbangkan untuk setiap proyek yang diusulkan. Nilai-nilai yang dihasilkan dievaluasi untuk menentukan proyek sistem mana yang akan menerima prioritas yang tertinggi.

**2.2.9.2 Analisis Sistem**

Menurut Kusrini (2007 : 40), tahapan analisis sistem dimulai karena adanya permintaan terhadap sistem baru. Permintaan bisa datang dari seorang Pimpinan/Manajer di luar departemen sistem informasi yang melihat adanya masalah atau menemukan adanya peluang baru. Namun, adakalanya inisiatif pengembangan sistem baru berasal dari bagian yang bertanggung jawab terhadap pengembangan sistem informasi. Tujuan utama dari analisis sistem adalah menentukan hal-hal secara detail yang akan dikerjakan oleh sistem yang diusulkan.

Dalam menganalisis sistem pendukung keputusan akan dilakukan langkah-langkah pembuatan model, yaitu :

1. Proses studi kelayakan yang terdiri dari penentuan sasaran, pencarian prosedur, pengumpulan data, identifikasi masalah, identifikasi kepemilikan masalah, hingga akhirnya terbentuk sebuah pernyataan masalah.
2. Proses perancangan model. Dalam tahapan ini akan diformulasikan model yang akan digunakan serta kriteria-kriteria yang ditentukan. Setelah itu, dicari alternatif model yang bias menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Berikutnya, tentukan variabel-variabel model. Setelah beberapa altenatif model diberikan, pada tahap ini akan ditentukan satu model yang akan digunakan dalam sistem pendukung keputusan yang akan dibangun.

Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem, adalah sebagai berikut :

1. *Identify,* mengidentifikasi (mengenal) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah dapat di definisikan sebagai suatu pertanyaan yang di inginkan untuk dipecahkan. Tahap identifikasi masalah sangat penting karena akan menentukan keberhasilan pada langkah-langkah selanjutnya.
2. *Understand,* adalah memahami kerja dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari secara terinci bagaimana sistem yang ada beroperasi. Untuk mempelajari operasi dari sistem ini diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.
3. *Analyze,* menganalisis sistem tanpa report.
4. *Report,* yaitu membuat laporan hasil analisis. Tujuan utama dari pembuatan laporan hasil analisis yaitu pelaporan bahwa analisis telah selesai dilakukan.

**2.2.9.3 Desain Sistem**

Dalam desain sistem, dibutuhkan alat bantu desain. Dalam tahapan ini, pengembang sistem bisa menentukan arsitektur sistemnya, merancang gambaran konseptual sistem, merancang database, perancangan interface, hingga membuat flowchart program. Salah satu alat bantu yang bisa digunakan dalam pembuatan sistem bantu keputusan adalah *Data Flow Diagram* (DFD). 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 antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

Menurut John Burch dan Gary Grudnitski, desain sistem dapat didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. (Jogiyanto, 2005 : 196)

Tahap desain sistem mempunyai dua tujuan utama :

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya.

Desain sistem dibagi dalam dua bagian, yaitu desain sistem secara umum *(general systems design)* dan desain sistem terinci *(detailed systems design).*

1. **Desain Sistem Secara Umum (*general systems design*)**

Pada tahap desain secara umum, komponen-komponen sistem informasi yang dirancang dengan tujuan dikomunikasikan kepada user bukan untuk pemrograman. Komponen sistem informasi yang di desain adalah model, output, input, database, teknologi, dan kontrol. (Jogiyanto, 2005 : 211)

a**.** Desain Model Secara Umum

Analisis sistem dapat mendesain model dari sistem informasi yang di usulkan dalam bentuk *physical* sistem dan *logical* model. Bagan alir sistem merupakan alat yang tepat digunakan untuk menggambarkan *physical systems,* logical model dapat digambar dengan diagram arus data. (Jogiyanto, 2005 : 211)

Bagan alir sistem merupakan bagan yang menunjukan arus pekerjaan secara keseluruhan dari sistem. Bagan alir sistem digambar dengan simbol-simbol sebagai berikut :

**Tabel 2.10** Daftar Simbol Bagan Alir Dokumen

| No. | Nama Simbol | Simbol | Keterangan |
| --- | --- | --- | --- |
|  | Terminal |  | Menunjukkan untuk memulai dan mengakhiri  Suatu proses |
|  | Dokumen |  | Menunjukkan dokumen input dan output baik itu proses manual, mekanik, atau computer |
|  | Kegiatan Manual |  | Menunjukan pekerjaan manual |
|  | Simpanan Offline | N  A  C | Menunjukkan file non-komputer yang diarsip urut angka (*numerical*), huruf (*alphabetical*), atau tanggal (*chronological*) |
|  | Proses |  | Menunjukkan kegiatan proses dari operasi program komputer |
|  | Operasi Luar |  | Menunjukkan operasi yang dilakukan diluar operasi computer |
|  | Hard Disk |  | Menunjukkan *input* dan *output* menggunakan *harddisk* |
|  | Keyboard |  | Menunjukkan *input* yang menggunakan *on-line keyboard* |
|  | Display |  | Menunjukkan *output* yang ditampilkan di monitor |
|  | Hubungan Komunikasi |  | Menunjukkan proses transmisi data melalui channel komunikasi |
|  | Garis Alir |  | Menunjukkan arus dari proses |
|  | Penjelasan |  | Menunjukkan penjelasan dari suatu proses |
|  | Penghubung |  | Menunjukkan penghubung ke halaman yang masih sama atau ke halaman yang lain |

(Sumber: Jogiyanto HM, 2005 : 802)

Untuk mempermudah penggambaran suatu sistem yang ada atau sistem yang baru yang akan dikembangkan secara logika dan tanpa memperhatikan lingkungan fisik data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan, maka digunakan Diagram Arus Data (DAD) atau *Data Flow Diagram* (DFD).

**Tabel 2.11** Daftar Simbol Diagram Alir Dokumen

| No | Simbol | Keterangan |
| --- | --- | --- |
|  |  | Simbol Proses, Menunjukan informasi dari masukan menjadi keluaran |
|  |  | Eksternal Entity, merupakan kesatuan dilingkungan luar system yang dapat berupa orang, organisasi atau system lain yang berada di lingkungan luarnya yang akan memberikan input seta menerima output dari system |
|  |  | Aliran atau arus data, menggambarkan gerakan paket data atau informasi dari suatu bagian kebagian yang lain, dimana penyimpanan mewakili lokasi penyimpana data |
|  |  | Penyimpanan, digunakan untuk memodelkan kumpulan data atau paket data |

(Sumber : Jogiyanto, 2005 : 700-807

b.Desain Output Secara Umum

Output adalah produk dari sistem informasi yang dapat dilihat. Output terdiri dari macam-macam jenis seperti hasil di media kertas, dan hasil di media lunak. Disamping itu output dapat berupa hasil dari suatu proses yang akan digunakan oleh proses lain dan tersimpan di suatu media seperti tape, disk, atau kartu. Yang dimaksud dengan output pada tahap desain ini adalah output yang berupa tampilan di media kertas atau di layar video. (Jogiyanto, 2005 : 213)

c. Desain Input Secara Umum

Alat input dapat digolongkan ke dalam 2 golongan, yaitu alat input langsung *(online input device)* dan alat input tidak langsung *(offline input device).* Alat input langsung merupakan alat input yang langsung dihubungkan dengan CPU, sedangkan alat input tidak langsung adalah alat input yang tidak langsung dihubungkan dengan CPU. (Jogiyanto, 2005 : 214)

d.Desain Database Secara Umum

Basis data (database) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diluar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. Sistem basis data adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi. (Jogiyanto, 2005 : 217)

**2. Desain Sistem Secara Rinci (*Detailed systems design***)

a.Desain Output Terinci

Desain output terinci dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk output-output dari sistem yang baru. Desain output terinci terbagi atas dua, yaitu desain output berbentuk laporan di media kertas dan desain output dalam bentuk dialog di layar terminal. (Jogiyanto, 2005 : 362)

1. Desain output dalam bentuk laporan : dimaksudkan untuk menghasilkan output dalam bentuk laporan dimedia kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk tabel dan berbentuk grafik atau bagan. (Jogiyanto, 2005 : 362)
2. Desain output dalam bentuk dialog layar terminal : merupakan rancang bangun dari percakapan antara pemakai sistem atau user dengan komputer. Percakapan ini dapat terdiri dari proses memasukkan data ke sistem, menampilkan output informasi kepada user, atau keduanya.

b.Desain Input Terinci

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data hasil dari transaksi merupakan masukan untuk sistem informasi. Hasil dari sistem informasi tidak lepas dari data yang dimasukan. Desain input terinci dimulai dari desain dokumen dasar sebagai penangkap input yang pertama kali. Jika dokumen dasar tidak di desain dengan baik, kemungkinan input yang tercatat dapat salah bahkan kurang. (Jogiyanto, 2005 : 375)

Fungsi dokumen dasar dalam penanganan arus data :

1. Dapat menunjukkan macam dari data yang harus dikumpulkan dan ditangkap.
2. Dapat dicatat dengan jelas, konsisten, dan akurat.
3. Dapat mendorong lengkapnya data, disebabkan data yang dibutuhkan disebutkan satu persatu di dalam dokumen dasarnya.

c.Desain Database Terinci

Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. Database merupakan salah satu komponen yang penting di dalam sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Penerapan database dalam sistem informasi disebut *database system.* (Jogiyanto, 2005 : 400)

**2.2.9.4 Seleksi Sistem**

Tahap ini merupakan tahap untuk memilih perangkat yang akan digunakan untuk sistem informasi. Pengetahuan dibutuhkan oleh pemilih sistem diantaranya adalah pengetahuan tentang siapa yang menyediakan teknologi ini, cara pemilikannya, dan sebagainya. Pemilihan sistem yang harus paham dengan teknik-teknik evaluasi untuk menyelesaikan sistem.

**2.2.9.5 Implementasi Sistem**

Menurut Kusrini (2007 : 43), Implementasi sistem merupakan tahapan untuk meletakkan sistem supaya siap untuk dioperasikan. Pada tahapan ini terdapat banyak aktifitas yang dilakukan, yaitu :

1. Pemrograman dan pengetesan program

Pemrograman merupakan kegiatan menulis program yang akan dieksekusi oleh komputer. Kode program harus berdasarkan dokumentasi yang disediakan oleh analis sistem hasil dari desain sistem.

1. Instalasi perangkat keras dan lunak

Proses pemasangan perangkat keras dan instalasi perangkat lunak yang sudah ada.

1. Pelatihan kepada pemakai

Manusia merupakan faktor yang diperlukan dalam sistem informasi. Jika ingin sukses dalam sistem informasi, maka personil-personil yang terlibat harus diberi pengertian dan pengetahuan tentang sistem informasi dan posisi serta tugas mereka.

1. Pembuatan dokumentasi

Dokumentasi adalah melakukan pencatatan terhadap setiap langkah pekerjaan pembuatan sebuah program yang dilakukan dari awal sampai selesai.

**2.2.9.6 Perawatan Sistem**

Perawatan sistem informasi adalah suatu upaya untuk memperbaiki, menjaga, menanggulangi, mengembangkan sistem yang ada. Perawatan ini di perlukan untuk meningkatkan efisiensi dan efektivitas kinerja sistem yang ada agar dalam penggunaannya dapat optimal. Beberapa alasan mengapa kita perlu memelihara sistem yang ada yaitu: agar dapat meningkatkan sistem / kinerja sistem, dan menyesuaikan dengan perkembangan, agar sistem yang ada tidak tertinggal.

Aplikasi yang professional dalam SDLC dan teknik maupun perangkat modeling yang mendukungnya adalah hal-hal keseluruhan yang terbaik yang dapat seseorang lakukan untuk meningkatkan maintainabilitas sistem.

Jenis-jenis perawatan sistem meliputi :

* 1. Perawatan korektif: adalah pemeliharaan yang mengkoreksi kesalahan – kesalahan yang ditemukan pada sistem, pada saat sistem di jalankan / berjalan.
  2. Pemeliharaan adaptif: yaitu pemelihaaan yang bertujuan untuk menyesuaikan perubahan yang terjadi.
  3. Pemeliharaan perfektif: pemeliharaan ini bertujuan untuk meningkatkan cara kerja suatu sistem.
  4. Pemeliharaan preventif: pemeliharaan ini bertujuan untuk menangani masalah-masalah yang ada.

**2.2.10 Teknik Pengujian Sistem**

Pengujian sistem adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean. Tujuan dari pengujian ini adalah diharapkan dengan minimal tenaga dan waktu untuk menemukan berbagai potensi kesalahan dan cacat. Harus didasarkan pada kebutuhan berbagai tahap pengembangan, desain dan dokumen lain atau program yang dirancang untuk menguji struktur internal, dan menggunakan contoh-contoh ini untuk menjalankan program untuk mendeteksi kesalahan.Pengujian sistem informasi harus mencakup pengujian perangkat lunak, pengujian perangkat keras dan pengujian jaringan. Pengujian hardware, jaringan pengujian berdasarkan indikator kinerja spesifik yang akan digunakan di sini pengujian lebih jauh adalah pengujian perangkat lunak.

**2.2.10.1 *White Box***

Pengujian *white-box (glass box)*, adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*. Dengan menggunakan metode pengujian *white-box*, perekayasa sistem dapat melakukan *test case* untuk memberikan jaminan bahwa :

1. Semua jalur independen pada suatu modul ditelusuri minimal 1 (satu) kali.
2. Semua jalur keputusan logis *True/False* dilalui.
3. Semua *loop* dieksekusi pada batas yang tercantum dan batas operasionalnya.
4. Struktur data internal digunakan agar validitas terjamin.

Pengujian *white-box* bisa dilakukan dengan pengujian *basis path*, metode ini merupakan salah satu teknik pengujian struktur kontrol untuk menjamin semua statemen dalam setiap jalur independen program dieksekusi minimal 1 kali dan tidak menjumpai *error message*. Perhitungan jalur independen dapat dilakukan melalui metrik *Cyclomatic Complexity.* Sebelum menghitung nilai *Cyclomatic Complexity, harus* diterjemahkan desain prosuderal ke grafik alir, kemudian dibuat *flow graphnya*, seperti pada gambar di bawah ini (Roger S. Pressman, 2002 : 536)



**Gambar 2.5** Contoh Bagan Alir

**Gambar 2.6** Contoh Grafik Alir

Keterangan :

1. *Node* adalah lingkaran yang merepresentasikan satu atau lebih statemen prosedural.
2. *Edge* adalah anak panah pada grafik alir.
3. *Region* adalah area yang membatasi edge dan node.
4. Simpul Predikat adalah simpul atau node yang berisi kondisi yang ditandai dengan dua atau lebih edge yang berasal darinya.

Darigambar *flowgraph* di atas didapat :

*Path* 1 =1– 11

*Path* 2 =1– 2 – 3 – 4 – 5 – 10– 1–11

*Path* 3 =1– 2 – 3 – 6 – 8 – 9 – 10– 1 – 11

*Path* 4 =1– 2 – 3 – 6 – 7 – 9–10–1–11

*Path* 1,2,3,4 yang telah didefinisikan diatas merupakan *basis set*untuk diagram alir.

*Cyclomatic complexity* digunakan untuk mencari jumlah *path*dalam satu *flowgraph*. Dapat dipergunakan rumusan sebagaiberikut :

1. Jumlah region grafikalir sesuai dengan *cyclomatic complexity*.

2. *Cyclomatix complexity* V(G) untuk grafik alir dihitung dengan rumus:

***V(G) =E– N +2*** …………………. (2.5)

Dimana :

E= jumlah *edge* pada grafik alir

N= jumlah *node* pada grafik alir

*Cyclomatix complexity* V(G) juga dapat dihitung dengan rumus :

***V(G) =P +1*** ………………….. (2.6)

Dimana P = jumlah *predicate node* pada grafik alir

Dari Gambar di atas dapat dihitung *cyclomatic complexity*:

1. *Flowgraph* mempunyai 4 region

2. V(G) = 11 *edge*– 9 *node* + 2 = 4

3. V(G) = 3 *predicatenode* + 1 = 4

Jadi *cyclomatic complexity* untuk *flowgraph* adalah 4.

**2.2.10.2 *Black Box***

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi tidak benar atau hilang.
2. Kesalahan antar muka.
3. Kesalahan pada struktur data (pengaksesan basis data).
4. Kesalahan inisialisasi dan akhir program.
5. Kesalahan performasi.

Pengujian ini berfokus pada persyaratan fungsional perangkat lunak dan merupakan komplemen dari pengujian *White-Box*. Hal ini dapat dicapai melalui :

1. Pengujian *Graph-based*: dimulai dengan membuat grafik sekumpulan node yang mempresentasikan objek (misal *New File*, Layar baru dengan atributnya), link (hubungan antar objek), *node-weight* (misal nilai data tertentu seperti atribut layar, perilaku), dan link-weight (karakteristik suatu link, misal menu select).
2. *Equivalence Partitioning*: membagi domain input untuk pengujian agar diperoleh kelas-kelas kesalahan (misal kelompok data karakter, atau atribut yang lain).
3. Analisis Nilai Batas: pengujian berdasarkan nilai batas domain input.
4. Pengujian Perbandingan: disebut juga pengujian *back-to-back* yang diterapkan pada pada suatu versi perangkat lunak atau perangkat lunak redun dan untuk memastikan konsistensinya
   1. **Perangkat Lunak Pendukung**

Adapun perangkat lunak pendukung yang digunakan oleh penulis dalam membangun sistem ini ada beberapa diantaranya adalah:

**Tabel 2.12** Perangkat Lunak Pendukung

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat Lunak Pendukung** | **Kegunaan** |
| 1. | Microsoft Visual Basic Net 2010 | Bahasa Pemrograman yang digunakan untuk membuat program. |
| 2. | Database MySQL | Sebuah perangkat lunak yang digunakan dalam pengoperasian basis data. |
|  | Crystall Report for Visual Studio | Digunakan untuk pembuatan laporan. |

**2.4 Kerangka Pikir**

1. Bagaimana merekayasa sebuah aplikasi data mining yang dapat mengklasifikasikan penyakit di suatu wilayah menggunakan algoritma *K-Nearest Neighbor*?
2. Bagaimana menerapkan hasil dari algoritma *K-Nearest Neighbor* dalam mengklasifikasi penyakit di suatu Wilayah?

**Masalah**

1. Pihak puskesmas kabila Membutuhkan aplikasi untuk mengklasifikasi penyakit di suatu wilayah.
2. Metode *K-Nearest Neighbor* dapat digunakan untuk mengklasifikasi penyakidi suatu wilayah.

**Peluang**

Membangun Aplikasi data mining untuk mengklasifikasikan Penyakit di suatu wilayah.

**Solusi**

1. Sistem Berjalan
2. Sistem Diusulkan

**Analisa Sistem**

1. Desain Model
2. Desain User Interface(Desain Output, Desain Input, Desain Menu Utama)
3. Desain Database
4. Desain Teknologi

**Desain Sistem**

1. Microsoft Visual Basic Net 2010
2. MySQL
3. Crystall Report for Visual Studio

**Pembangunan Sistem**

1. White Box.
2. Black Box.
3. Confusion Matrix

**Pengujian**

Puskesmas Kabila

**Implementasi**

1. Merekayasa aplikasi data mining yang dapat mengklasifikasi penyakit di suatu wilayah menggunakan algoritma *K-Nearest Neighbor*.
2. Mengetahui hasil penerapkan dari algoritma *K-Nearest Neighbor* dalam mengklasifikasi penyakit di suatu Wilayah**.**

**Tujuan**

**Gambar 2.7** Kerangka Pikir