

BAB 2

LANDASAN TEORI

2.1 Surat

Surat menurut Thomas Wiyasa Bratawidjaja adalah alat untuk menyampaikan sesuatu maksud secara tertulis dari pihak yang satu ke pihak yang lain. Atau dengan kata lain surat menyurat merupakan salah satu kegiatan berbahasa yang dilakukan dalam komunikasi tertulis[12].

Jenis surat dibagi tiga yaitu[12] :

a. Surat Pribadi

Surat pribadi adalah surat yang ditujukan kepada perorangan atau pribadi yang biasanya berisi hal-hal yang menyangkut hubungan pribadi dan pengirimnya adalah seorang teman atau keluarga. Bahasa yang digunakan tidak begitu formal dan cenderung akrab, santai dan tidak tergantung pada kaidah yang berlaku pada surat dinas maupun surat niaga.

b. Surat Dinas / Surat Resmi

Surat Dinas (*official letter*) surat resmi yang digunakan dalam aktifitas perkantoran baik seseorang maupun dari satu kantor ke kantor lainnya. Bahasa yang digunakan bersifat lugas, langsung pada permasalahan dan membicarakan hal-hal yang penting saja. Pada surat dinas biasanya ada nomor surat, lampiran dan perihal.

c. Surat Bisnis / Niaga (business letter)

Surat bisnis/niaga adalah surat yang digunakan dalam kegiatan niaga atau perdagangan atau perusahaan. Surat ini dapat berasal dari perusahaan kepada pribadi/perorangan dan sebaliknya atau dari perusahaan kepada instansi pemerintah/swasta. Contoh surat permintaan penawaran, surat penawaran, surat pemesanan, surat pengiriman, surat tuntutan atau klaim, faktur, kuitansi dan lain lain.

Sedangkan surat masuk adalah semua surat dinas yang diterima[6]. Surat yang diterima tersebut baik dari instansi pemerintah maupun instansi non-

pemerintah. Dalam menyampaikan informasi surat dinas tersebut memiliki beberapa aturan dan objek seperti yang dijelaskan pada poin 2.1.1. Aturan Surat Dinas Resmi

Aturan surat dinas resmi yang diatur dalam Pedoman Tata Naskah Dinas Instansi Pemerintah Pedoman Tata Naskah Dinas Instansi Pemerintah adalah sebagai berikut [2]:

- a. Jenis naskah dinas korespondensi eksternal hanya ada satu macam, yaitu surat dinas.
- b. Pengertian Surat dinas adalah naskah dinas pelaksanaan tugas pejabat dalam menyampaikan informasi kedinasan berupa pemberitahuan, pernyataan, permintaan, penyampaian naskah dinas atau barang, atau hal kedinasan lainnya kepada pihak lain di luar instansi/organisasi yang bersangkutan.
- c. Wewenang Penandatanganan Surat dinas ditandatangani oleh pejabat sesuai dengan tugas, fungsi, wewenang, dan tanggung jawabnya.

2.1.1 Objek atau Susunan Surat Dinas

Objek atau susunan surat dinas adalah ketentuan isi dari surat dinas resmi yang sesuai dengan Pedoman Tata Naskah Dinas Instansi Pemerintah, dikeluarkan oleh Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia Tahun 2012. Dapat dilihat pada Gambar 2.1 contoh susunan surat dinas resmi yang akan digunakan[2].

	NAMA INSTANSI JALAN TELEPON FAKSIMILE	Kop surat yang berupa logo, nama instansi dan alamat lengkap yang telah dicetak
Nomor : ..J.J.J.J.. Sifat : Lampiran : Hal :	(Tgl., Bln., Thn)	
Yth. (Alinea Pembuka) (Alinea Isi) (Alinea Penutup)	Alamat tujuan yang ditulis di bagian kiri
Tembusan: 1. 2. 3.	Nama Jabatan, (Tanda Tangan dan Cap Instansi) Nama Lengkap	Nama jabatan dan nama lengkap yang ditulis dengan huruf awal capital

Gambar 2.1 Naskah Surat Dinas Resmi [2]

Susunan surat dinas resmi sesuai 1 adalah sebagai berikut [2]:

- a. Kepala surat, bagian kepala surat dinas terdiri dari
 - 1) Kop surat dinas, yang berisi lambang negara dan nama jabatan (untuk pejabat negara) atau logo dan nama instansi (untuk nonpejabat negara) secara simetris;
 - 2) Nomor, sifat, lampiran, dan hal, diketik dengan huruf awal kapital di sebelah kiri di bawah kop surat dinas;
 - 3) Tempat dan tanggal pembuatan surat, diketik di sebelah kanan atas sejajar/sebaris dengan nomor;

- 4) Kata Yth., ditulis di bawah hal, diikuti dengan nama jabatan yang dikirim surat; e) alamat surat, ditulis di bawah Yth.
- b. Batang tubuh surat, bagian batang tubuh surat dinas terdiri dari alinea pembuka, isi, dan penutup.
- c. Kaki surat, bagian kaki surat dinas terdiri dari
 - 1) Nama jabatan, ditulis dengan huruf awal kapital, diakhiri tanda baca koma;
 - 2) Tanda tangan pejabat;
 - 3) Nama lengkap pejabat/penanda tangan, ditulis dengan huruf awal kapital;
 - 4) Stempel/cap dinas, yang digunakan sesuai dengan ketentuan;
 - 5) Tembusan, yang memuat nama jabatan pejabat penerima (jika ada)

2.2 Text Mining

Text mining adalah salah satu variasi dari ruang lingkup *data mining*, yang mencoba untuk mencari pola-pola yang unik dari *database* yang besar. *Text mining* merujuk kepada proses dari mengekstraksi hal-hal yang diinginkan dan informasi serta pengetahuan yang belum terungkap dari teks yang tidak terstruktur[13].

Text mining adalah prosedur untuk mengambil informasi dengan menganalisis hubungan, pola-pola, dan aturan-aturan dari data teks. Salah satu gagasan utama dari *Text mining* adalah mentransformasikan teks menjadi data numerik. Metode ini akan mentransformasikan data dari teks menjadi bentuk numerik standar dengan cara membentuk format *spreadsheet* dari teks atau bahasa sederhananya tidak terstruktur menjadi terstruktur[1]. Teks yang sudah ditransformasikan kemudian akan dilakukan analisis data seperti proses *data mining* yang kebanyakan menggunakan *machine learning*. Dalam *machine learning* terdapat *supervised learning* dan *unsupervised learning*. Perbedaan keduanya adalah jika *supervised learning* memiliki data latih sebelumnya untuk dapat menentukan kelas yang akan diujikan selanjutnya, sementara *unsupervised*

learning tidak menggunakan data latih untuk mengelompokan kelas yang akan diujikan selanjutnya.

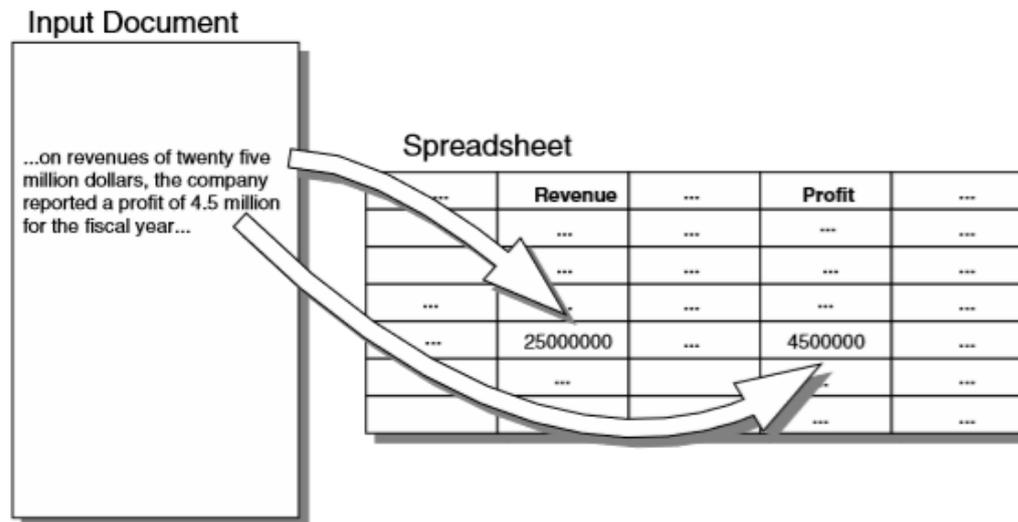
Kemudian perbedaan dari *data mining* pada umumnya dengan *text mining* adalah, didalam *text mining* pola-polanya di ekstrak dari bahasa alami (*natural language*) bukan dari data yang sudah terstruktur[13]. Jadi, sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Kesamaan dari *text mining* dan *data mining* terletak setelah proses transformasi kumpulan data numerik dari dokumen yang tidak terstruktur, keduanya sama-sama menggunakan metode statistik dan *machine learning* yang sudah ada untuk memproses informasi[13].

Dari informasi yang didapat dari *text mining* dokumen surat masuk inilah akan disusun menjadi informasi yang terstruktur. Dibutuhkannya tahapan proses dalam *text mining* yang tepat agar informasi yang dibutuhkan tetap terjaga. Dalam penelitian ini *text mining* yang dilakukan meliputi beberapa proses, yaitu *text preprocessing*, *weighting* (pembobotan), dan klasifikasi.

2.2.1 Ekstraksi Informasi

Ekstraksi informasi adalah cabang ruang lingkup dari *text mining*. Yang bertujuan untuk mengubah hasil proses *text mining* menjadi akar yang sama dengan dunia data yang terstruktur dalam *data mining*[1]. Metode Ekstraksi Informasi memungkinkan mengambil informasi aktual yang ada dalam teks bukan hanya kategorisasi *tag* teks secara umum. Dengan demikian, teknik *preprocessing* yang melibatkan Ekstraksi Informasi cenderung akan menciptakan model yang lebih optimal dan fleksibel untuk dokumen dalam sistem *text mining*[14].

Sederhananya proses ekstraksi informasi ini mengecek dokumen dan mengisikannya kedalam sel, proses yang sama seperti membuat Tabel dalam *database*[1]. Ekstraksi informasi pada penelitian ini akan dilakukan pada dokumen surat masuk. Gambaran ekstraksi informasi dapat dilihat pada Gambar 2.2



Gambar 2.2 Contoh Proses Ekstraksi Informasi Dokumen [1]

Pada Gambar 2.2 adalah contoh dari ekstraksi informasi dimana dari sebuah teks masukan diambil informasi pendapatan (*revenue*) sebesar 2,5 juta dan keuntungan sebesar 4,5 juta yang kemudian disusun dalam tabel (*spreadsheet*).

2.2.2 *Converting*

Converting adalah tahap menyiapkan dokumen sebelum dilakukannya proses *preprocessing*. *Converting* adalah merubah suatu ekstensi file menjadi ekstensi baru, dan mengikuti format dari ekstensi baru tersebut. Dalam penelitian ini yang akan dirubah adalah dokumen surat yang pada kondisi awalnya berekstensi .pdf menjadi file berekstensi baru yaitu .xml. Tujuan dari *converting* ini adalah untuk menyiapkan dokumen yang lebih mudah dibaca sebagai teks.

Dalam penelitian ini akan menggunakan alat melalui API dari website <https://pdftables.com>. Keterangan dari masing-masing ekstensi adalah :

- a. Ekstensi file awal (.pdf) : Standar format file yang umum digunakan untuk menyimpan dokumen digital adalah format file PDF yang dikembangkan oleh perusahaan Adobe sejak tahun 1990 hingga saat ini. [15]. Alat yang kita gunakan untuk mendigitalkan kertas dokumen kita adalah *scanner*, saat ini ada beberapa jenis teknologi *scanner* yang bisa kita temukan di pasar, seperti *flatbed scanner*, *automatic document feeder (ADF) scanner*,

portable scanner, book scanner, yang masing-masing memiliki fungsi yang berbeda-beda[15].

- b. Ekstensi file baru (.xml) : *Extenable Markup Language* (XML) sebenarnya bukan sebuah bahasa pemrograman, karena XML adalah bahasa *mark up*. XML digunakan untuk *mark up* (penanda) terhadap suatu dokumen teks. Simbol mark up yang digunakan oleh XML ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>). Kedua tanda ini disebut tag. *Tag* yang digunakan sebagai tanda penutup diberi karakter garis miring (</..>)[16].

2.2.3 *Text Preprocessing*

Text preprocessing adalah tahapan untuk mempersiapkan teks menjadi data yang akan diolah di tahapan berikutnya. Inputan awal pada proses ini adalah berupa dokumen[14]. Dalam prosesnya *preprocessing* yang digunakan dalam penelitian ini memiliki 3 tahap, yaitu *filtering*, *tangging*, dan *tokenization*. Sebelum melalui 3 tahap tersebut dokumen harus melalui proses *converting* untuk menyiapkan dokumen yang lebih mudah dibaca sebagai teks.

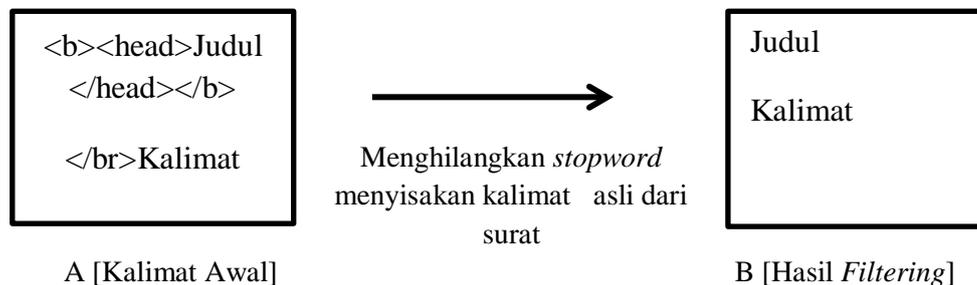
2.2.3.1 *Filtering*

Filtering merupakan proses penghilangan *stopword*. *Stopword* adalah kata-kata yang sering kali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Didalam bahasa Indonesia *stopword* dapat disebut sebagai kata tidak penting, misalnya “di”, ”oleh”, “pada”, ”sebuah”, ”karena” dan lain sebagainya[17]. Contoh umum tahap *filtering* dapat dilihat pada Gambar 2.3 dimana yang akan dihilangkan dari bagian A kalimat awal adalah kata sambung dalam B. Indonesia yaitu “akan”, “di-”, ”kepada”, dan “yang”. Hasilnya dapat dilihat pada bagian B Gambar 2.3 dimana *stopword* tadi sudah dihilangkan.



Gambar 2.3 Contoh Umum Filtering

Dalam penelitian ini yang akan menjadi *stopword* adalah *tag* dari html dan kata-kata yang ingin dipertahankan adalah isi dari dokumen surat tersebut. Contoh *filtering* yang digunakan dalam penelitian ini dapat dilihat pada Gambar 2.4 dimana yang akan dihilangkan dari bagian A Kalimat Awal adalah tanda dari HTML seperti “”, “<head>”, dan “
”. Dapat dilihat hasil penghilangan tanda HTML tersebut pada bagian B Gambar 2.4.

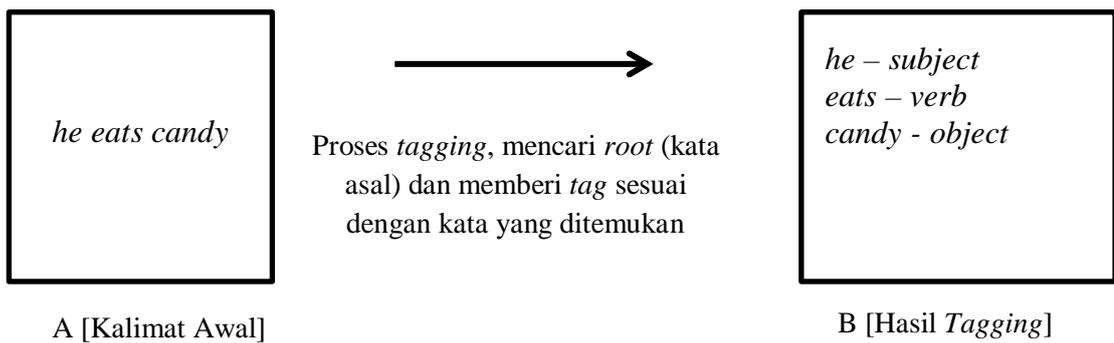


Gambar 2.4 Contoh Filtering dalam Penelitian

2.2.3.2 Tagging

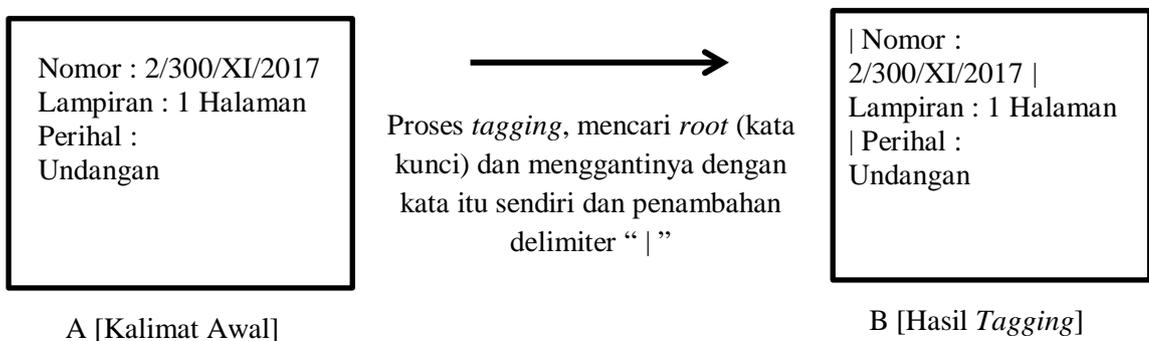
Tagging adalah membagi kata ke dalam kategori-kategori berdasarkan fungsi dari kalimat yang terbentuk dari kata tersebut[14]. Tujuan pemberian *Tag* adalah untuk menentukan kategori dari kemungkinan-kemungkinan kategori yang terdapat dalam contoh teks tersebut[1]. Contoh penggunaan *tagging* yang paling umum adalah dalam B. Inggris dimana terdapat 5 sampai 7 kategori, dengan kebanyakan memiliki subjek, kata kerja, kata sifat, kata benda, penunjukan, dan kata penghubung. Contoh umum tahap *tagging* dapat dilihat pada Gambar 2.5

dimana bagian A kalimat awal akan diberikan tanda dalam B. Inggris apabila ditemukan kesesuaian dengan kata kunci. Pada bagian B Gambar 2.5 adalah hasil dari pemberian *tag* dimana token “*he*” sesuai dengan *root* yang memiliki *tag subject* sehingga diberikan tanda *subject*, token “*eats*” sesuai dengan *root* yang memiliki *tag verb* sehingga diberikan tanda *verb*, dan seterusnya.



Gambar 2.5 Contoh Umum *Tagging*

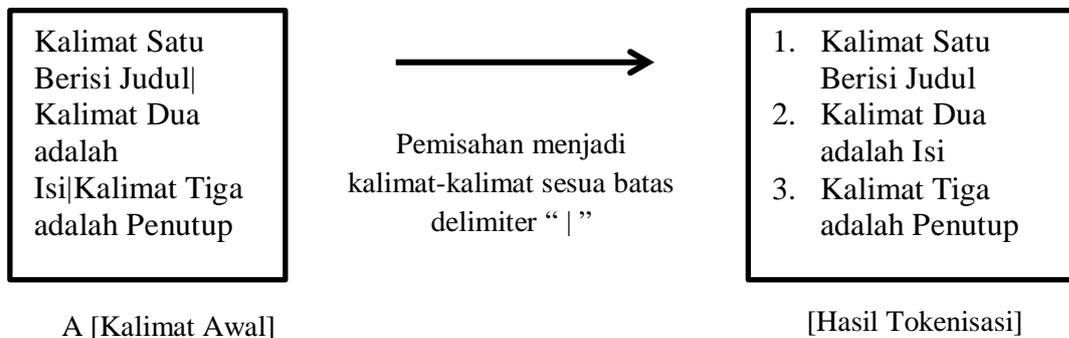
Dasar *tagging* ini digunakan dalam penelitian ini untuk menjadi salah satu metode *preprocessing text*, untuk mencari kata-kata kunci yang kemudian akan diberi *tag* khusus. Dalam penelitian ini, dapat dilihat proses *tagging* seperti di Gambar 2.6 dimana bagian A kalimat awal akan diberikan tanda apabila ditemukan kesesuaian dengan kata kunci. Pada bagian B Gambar 2.6 adalah hasil dari pemberian tanda dimana ditemukan kata “Nomor”, “Lampiran” dan “Perihal” yang bersesuaian dengan kata kunci sehingga diberikan tanda “|”.



Gambar 2.6 Contoh *Tagging* dalam Penelitian

2.2.3.3 Tokenization (Tokenisasi)

Tokenisasi bertujuan untuk memecah karakter yang kontinyu (teks yang panjang) menjadi bagian-bagian yang lebih bermakna atau detail. Proses ini dapat terjadi dalam beberapa tingkatan. Dokumen dapat dipecah menjadi bab, bagian, paragraf, kalimat, kata, bahkan suku kata dan fonem. Pendekatan proses ini yang paling sering ditemukan dalam sistem *text mining* adalah memecah teks menjadi kalimat atau kata-kata, yang disebut tokenisasi[14]. Proses tokenisasi dapat dilihat Pada Gambar 2.7 dimana setiap ditemukan tanda “|” pada bagian A Kalimat Awal Akan dipisahkan menjadi kalimat-kalimat yang terpisah pada bagian B Gambar 2.7.



Gambar 2.7 Contoh Pemecahan Kalimat (Tokenisasi)

2.2.4 Pembobotan

Pembobotan menggunakan teknik *information retrieval* (mendapatkan informasi). Secara garis besar, metode pengambilan informasi terbagi dalam dua kategori, yaitu *document selection* dan *document ranking*[18]. Dalam *document selection*, kueri dianggap sebagai batasan spesifik untuk memilih dokumen/informasi yang relevan. Metode khas dari kategori ini adalah *boolean retrieval model*, di mana pengguna menginisialisasi ekspresi kata kunci *Boolean*, seperti "mobil dan bengkel," "teh atau kopi," atau "sistem *database* tetapi tidak oracle." Sistem pengambilan informasi akan mengambil dan mengembalikan nilai dokumen yang memenuhi ekspresi *boolean* tersebut dengan nilai 1 dan selebihnya adalah 0.

Sederhananya bila memenuhi kueri fungsi bobot *boolean* akan bernilai 1 dan selain itu bernilai 0. Nilai fitur yang digunakan dalam pembobotan diambil

dari penelitian Aditya Iftikar Riaddy, dkk[8] dan fitur panjang kalimat dari penelitian Novriana[9], yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Fitur

Nama fitur	Deskripsi
Fitur Lokal	
INITCAPS	Dimulai dengan huruf kapital
ALLCAPS	Seluruh karakter adalah huruf kapital
CONTAINSDIGIT	Mengandung digit angka
ALLDIGITS	Seluruh karakter adalah digit angka
PHONEORZIP	Nomor telepon atau kode pos
CONTAINSDOTS	Mengandung paling sedikit satu titik
CONTAINSDASH	Mengandung paling sedikit satu garis
ACRO	Akronim / singkatan
LONELYINITIAL	Inisial seperti A.
SINGLECHAR	Hanya mengandung satu karakter
CAPLETTER	Hanya mengandung satu huruf kapital
PUNC	Tanda baca
URL	Alamat URL
EMAIL	Alamat <i>email</i>
Fitur Named Entity	
LOCATION	Nama lokasi
PERSON	Nama orang
ORGANIZATION	Nama organisasi
MONEY	Nominal uang
PERCENT	Nominal dalam persen
DATE	Waktu dalam penanggalan
TIME	Menunjukkan Jam
CHAR_LENGTH	Menghitung jumlah karakter

Sementara fitur yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2.2

Tabel 2.2 Fitur yang Digunakan

Nama fitur	Deskripsi
Fitur Lokal	
ALLCAPS	Seluruh karakter adalah huruf kapital
CONTAINSDIGIT	Mengandung digit angka
CONTAINSDASH	Mengandung paling sedikit satu garis

PUNC	Tanda baca (Koma)
EMAIL	Alamat <i>email</i>
Fitur <i>Named Entity</i>	
LOCATION	Nama lokasi
DATE	Waktu dalam penanggalan
Fitur Panjang Karakter	
CHAR_LENGTH	Menghitung jumlah karakter

Tambahan fitur yang dikembangkan dari fitur yang sudah ada pada Tabel 2.3

Tabel 2.3 Fitur Pengembangan

Nama fitur	Deskripsi	Pengembangan Dari
CONTAINSCOLON	Mengandung paling sedikit 1 titik dua	CONTAINSDASH
CONTAINS2SLASH	Mengandung paling sedikit 2 garis	CONTAINSDASH
INITIAL_KEYWORD	Tanda baca (Koma)	ACRO dan LONELYINITIAL

2.2.5 Klasifikasi Teks

Klasifikasi teks adalah pengelompokan atau pemberian label pada dokumen sesuai dengan kategori yang sesuai dengan konten dokumen tersebut. Kumpulan kategori tersebut sering disebut dengan aturan kosakata (*Controlled Vocabulary*). Klasifikasi teks adalah aksi untuk membagi suatu kumpulan dokumen menjadi dua atau lebih kelas yang setiap dokumen dapat memiliki satu kelas atau kelas ganda[13]. Klasifikasi teks menggunakan *machine learning* sebagai media pembelajaran komputer untuk dapat mengklasifikasi teks.

Beberapa metode yang dapat digunakan untuk mengklasifikasikan teks misalnya *Naïve Bayes*, *Support Vector Machine*, *K-Nearest Neighbour*, dll. Dalam penelitian ini metode yang digunakan adalah *Fuzzy K-Nearest Neighbour* yang adalah pengembangan dari Algoritma *K-Nearest Neighbour*.

2.3 Logika Fuzzy

Logika *fuzzy* pertama kali dikembangkan pada tahun 1965 oleh L.A. Zadeh mengenai teori himpunan *fuzzy*. Logika *fuzzy* umumnya diterapkan pada masalah-masalah yang mengandung unsur samar (ketidakjelasan). Berfungsi sebagai penghubung bahasa mesin (*assembly*) yang presisi dengan bahasa alami

yang lebih menekankan pada makna. Logika *fuzzy* dikembangkan berdasarkan bahasa alami (*natural language*)[19].

Logika *fuzzy* memperhitungkan nilai diantara nilai *boolean* (0 dan 1). Sehingga setiap kemungkinan kejadian yang terjadi didalam range 0 dan 1 akan tetap diperhitungkan. Misalnya adalah jika tinggi seseorang 1,8m dikatakan “tinggi”, bagaimana dengan orang yang tinggi badannya 1,75 atau bahkan 1,79m?. Itulah yang dikatakan sebagai logika *fuzzy* ketika menerjemahkan nilai “Kurang lebih tinggi” atau “Agak tinggi” kedalam nilai kebenaran (diantara 0 dan 1).

2.4 Himpunan *Fuzzy* dan Nilai Keanggotaan

Dalam himpunan konvensional atau biner hanya terdapat dua nilai yaitu 0 dan 1, jika himpunan semesta diberi notasi berikut pada rumus 2.1 notasi untuk himpunan konvensional[20].

$$u_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (2.1)$$

Apabila x elemen A maka bernilai 1, jika tidak maka bernilai 0. Berbeda dengan himpunan *fuzzy* yang menggunakan fungsi karakteristik menjadi nilai keanggotaan. Contohnya adalah himpunan *fuzzy* untuk bilangan riil yang lebih besar dari 0[20]. Dapat dilihat pada rumus 2.2

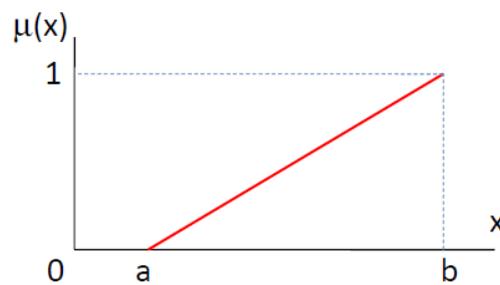
$$u(x) = \begin{cases} x^2/(x^2 + 1), & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.2)$$

Bilangan yang tidak lebih besar dari 0 maka tidak akan masuk ke set $u(x)$ tetapi langsung masuk ke nilai $u(0) =$ bernilai 0, Sementara bilangan yang lebih besar dari 0 akan dihitung nilai keanggotaan sesuai dengan seberapa lebih besar dari nilai 0. Dorongan di balik pengenalan teori himpunan *fuzzy* adalah untuk menyediakan sarana untuk mendefinisikan kategori yang secara inheren tidak tepat.

Keuntungan yang disediakan oleh himpunan adalah bahwa derajat keanggotaan dalam satu himpunan dapat ditentukan, bukan hanya biner. Dengan kata lain himpunan *fuzzy* akan menjelaskan secara detail apa tingkatan objek pada setiap kelasnya.

Selanjutnya adalah jumlah nilai keanggotaan *fuzzy* harus dapat dilacak secara matematis. Misalnya nilai keanggotaan *fuzzy* dalam 2 kelas, nilai keanggotaan yang bukan bernilai 0,5 maka menunjukkan lebih berat ke salah satu kelas, nilai 0,5 tadi dapat disebut “*bounding region*” atau pemisah 1 kelas dengan kelas lainnya. Untuk mencari nilai keanggotaan *fuzzy* dapat dilakukan dengan beberapa cara, diantaranya adalah[19] :

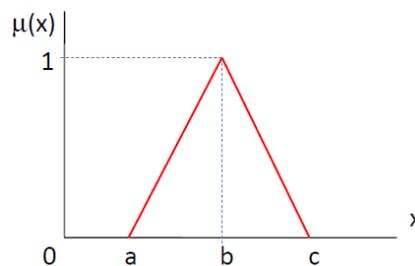
a. Fungsi Linear



Gambar 2.8 Grafik Fungsi Linear Nilai Keanggotaan *Fuzzy*

$$\mu(x) = \begin{cases} 0; & x \leq a \\ (x - a) / (b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.3)$$

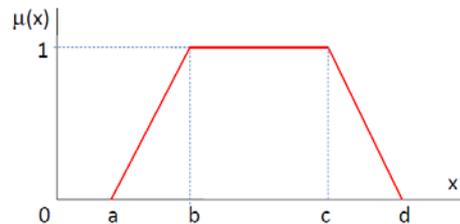
b. Fungsi Segitiga



Gambar 2.9 Grafik Fungsi Segitiga Nilai Keanggotaan *Fuzzy*

$$\mu(x) = \begin{cases} 1; & b \leq x \leq c \\ (b - x) / (c - b); & a \leq x \leq b \\ (d - x) / (d - c); & x \geq b \end{cases} \quad (2.4)$$

c. Fungsi Trapesium



Gambar 2.10 Grafik Fungsi Linear Nilai Keanggotaan Fuzzy

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x - a) / (b - a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d - x) / (d - c); & x \geq d \end{cases} \quad (2.5)$$

2.5 Algoritma *K-Nearest Neighbour* (K-NN)

K-Nearest Neighbour tidak memerlukan proses *preprocessing* sampel berlabel yang ditetapkan sebelum digunakan. Klasifikasi K-NN berdasarkan vektor sampel input y , yang mana dari klasifikasi tidak diketahui, menjadi ke kelas klasifikasi data tetangga terdekatnya[20].

Cara kerja K-NN adalah mencari jarak data input dengan setiap data yang sudah ada (data latih) dan kemudian ditentukan klasifikasi dengan mengambil kelas yang sama dengan data yang memiliki jarak tetangga terdekat, Nilai k adalah untuk menjadi pembatas jumlah nilai jarak yang akan dibandingkan dengan menggunakan fungsi min. Tahapan K-NN adalah tentukan nilai k , kemudian hitung jarak antara data testing ke data uji, urutkan jarak berdasarkan jarak terkecil/terdekat, tentukan nilai kelas data testing berdasarkan mayoritas dalam jarak sejumlah k .

2.6 Algoritma *Fuzzy K-Nearest Neighbour* (Fuzzy K-NN)

Algoritma *Fuzzy K-Nearest Neighbour* memberikan nilai membership sebagai fungsi pola jarak dari sejumlah himpunan *K-Nearest Neighbour* dan pemberian nilai keanggotaan *neighbour* pada kelas tertentu. Sehingga pada algoritma ini data *testing* yang akan diklasifikasikan akan memiliki nilai keanggotaan pada semua kelas. Klasifikasi algoritma ini nantinya akan memilih

nilai keanggotaan kelas pada data *testing* yang paling tinggi[20]. Proses pertama adalah menentukan jarak dari setiap data uji dengan data latih. Berikut merupakan persamaan untuk memberikan nilai jarak secara umum dengan menggunakan metode *euclidean distance*, dapat dilihat pada rumus 2.6.

$$ED_{i,h} = \sqrt{\sum_{j=1}^p (a_{i,j} - a_{h,j})^2} \quad (2.6)$$

Keterangan :

$ED_{i,h}$ = Jarak *euclidean* data i dengan data h

p = Nilai maks. dari indeks bobot yang dihitung

j = Indeks bobot yang dihitung

$a_{i,j}$ = Nilai bobot indeks ke-j di data i

$a_{h,j}$ = Nilai bobot indeks ke-j di data h

Dari rumus 2.6 akan dilakukan penyesuaian variabel dalam penelitian ini, hasil penyesuaian dapat dilihat pada rumus 2.7

$$\|x - x_j\| = \sqrt{\sum_{b=1}^p (x_b - x_{j,b})^2} \quad (2.7)$$

Keterangan :

$\|x - x_j\|$ = Jarak *euclidean* data uji x dengan data latih ke-j

p = Nilai maks. dari indeks bobot yang dihitung

b = Indeks bobot yang dihitung

x_b = Nilai bobot indeks ke-b di data uji x

$x_{j,b}$ = Nilai bobot indeks ke-b di data latih x_j

Untuk perhitungan nilai keanggotaan kelas data uji menggunakan rumus 2.8

$$\mu_i(x) = \frac{\sum_{j=1}^k \mu_{ij} \left(\frac{1}{\|x - x_j\|^{\frac{2}{m-1}}} \right)}{\sum_{j=1}^k \left(1/\|x - x_j\|^{2/(m-1)} \right)} \quad (2.8)$$

Keterangan :

$\mu_i(x)$ = Nilai keanggotaan kelas ke-i pada data uji x.

μ_{ij} = Nilai keanggotaan tetangga k (dari data latih x_j) pada kelas ke-i
(bernilai 1 jika kelas tetangga k dari data latih x_j sama dengan kelas ke-i,
dan bernilai 0 jika tidak)[23]

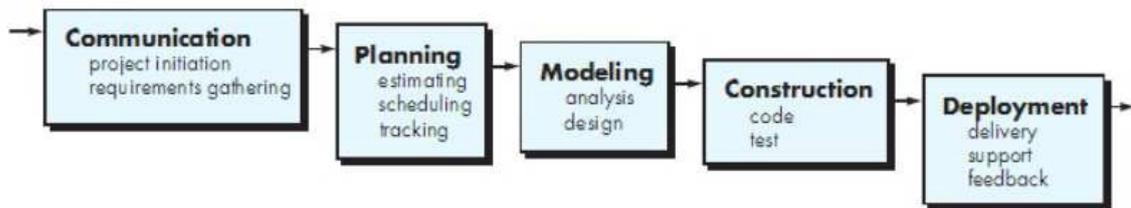
j = Indeks data latih

k = Banyaknya nilai ketetanggaan terdekat yang diambil

m = Berat pangkat bobot yang sebanding dengan jarak antara x dan x_j

2.7 Model linear/Waterfall

Menurut Pressman[11] model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan, oleh karena itu disebut seperti air terjun atau *waterfall*. Fase-fase dalam *Waterfall Model* menurut dapat dilihat pada Gambar 2.11.



Gambar 2.11 Tahapan *Waterfall Pressman* [11]

Untuk penjelasan setiap tahapnya adalah berikut :

a. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

b. *Planning (Estimating, Scheduling, Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko- resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem.

c. *Modeling (Analysis & Design)*

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami an besar dari apa yang akan dikerjakan.

d. *Construction (Code & Test)*

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem

dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

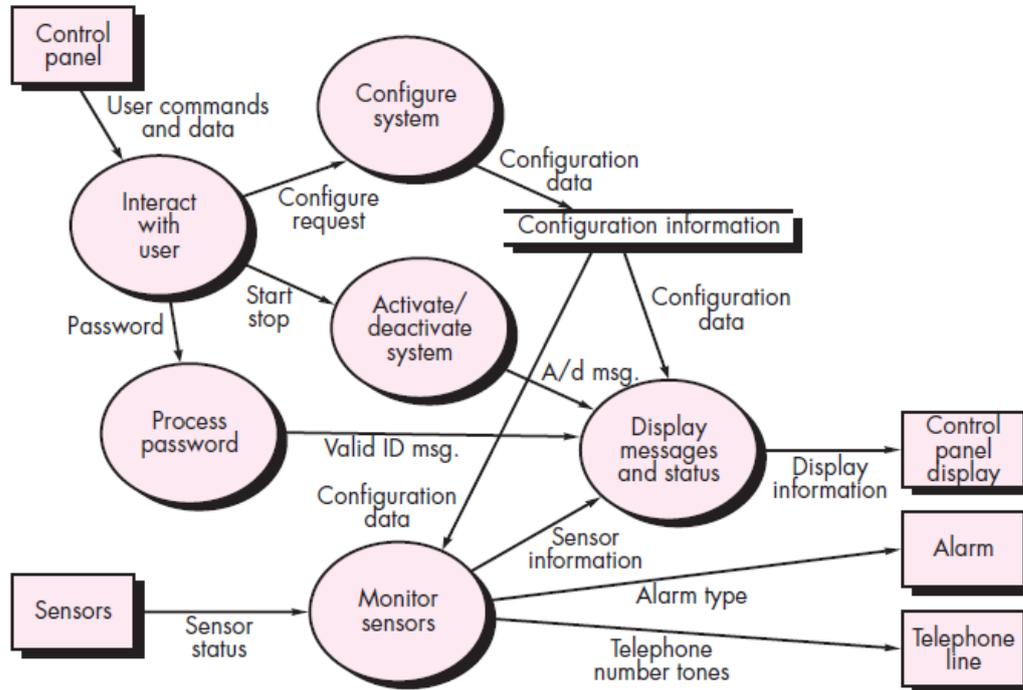
e. Deployment (*Delivery, Support, Feedback*)

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya.

2.8 Pemrograman Terstruktur

Model pemrograman terstruktur atau dapat disebut analisis terstruktur adalah menganggap data dan proses yang mengubah data sebagai entitas yang terpisah[11]. Objek data dimodelkan dengan cara yang mendefinisikan atribut dan hubungan mereka. Proses itu memanipulasi objek data dimodelkan dengan cara yang menunjukkan bagaimana mereka mentransformasi data sebagai objek data mengalir melalui sistem[11].

Data flow diagram (DFD) memungkinkan pengguna untuk mengembangkan model dari ranah informasi dan ranah fungsional. DFD mengkan alur data yang bergerak melalui proses mewakili perangkat lunak yang dibangun[11]. Contoh *data flow diagram* (DFD) dapat dilihat pada Gambar 2.12.

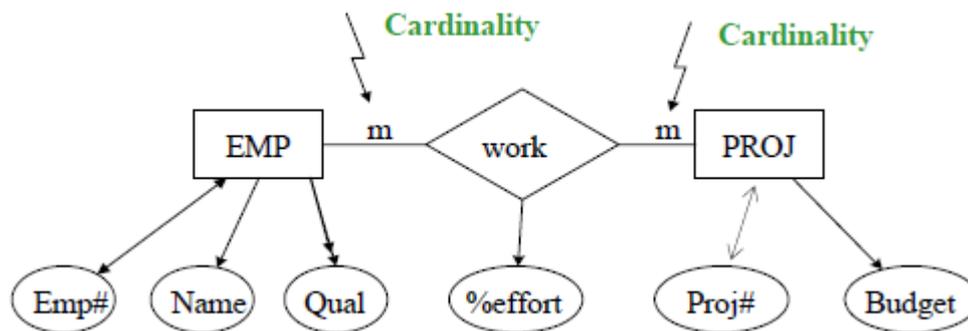


Gambar 2.12 Contoh DFD Level [11]

Gambar 2.12 adalah contoh dari *data flow diagram* Level 1 pada sistem keamanan rumah. Dimana digambarkan keterkaitan setiap entitas, data dan proses data yang dilakukan seperti misalnya entitas sensor memberikan data status sensor ke proses *monitor sensors* (pengawasan sensor).

2.9 Entity Relationship Diagram (ERD)

Model data ERD (*Entity Relationship Diagram*) merupakan diagram yang menggambarkan skema hubungan antar entitas atau objek[21]. Dengan ERD dapat dikatan dan dipetakan bagaimana table-Tabel saling berelasi. Dengan adanya ERD sebagai dasar maka *database* yang sudah ada dapat dikembangkan menjadi data yang dapat diolah ke dalam aplikasi. Relasi – relasi data yang sudah digambarkan pada ERD akan diimplementasikan ke *database* dalam aplikasi tersebut sehingga *database* aplikasi tersebut akan lebih terorganisir. Dalam Gambar 2.13 adalah contoh model ERD



Gambar 2.13 Contoh Umum ERD [21]

Gambar 2.13 adalah contoh dari ERD pada suatu gambaran proyek atau pekerjaan. Dimana ada entitas EMP atau Karyawan yang memiliki atribut id, nama, dan kualitas yang memiliki hubungan *work* atau mengerjakan dengan entitas PROJ atau proyek yang memiliki atribut id dan anggaran. Sementara *cardinality* menunjukkan jenis hubungan dari entitas EMP dan entitas PROJ.

2.10 PHP (*Hypertext Preprocessor*)

HP (*Hypertext Preprocessor*) merupakan bahasa *scripting* yang tergabung menjadi satu dengan HTML dan dijalankan pada *server side* atau semua perintah yang diberikan akan secara penuh dijalankan pada server, sedangkan yang dikirimkan ke klien (browser) hanya berupa hasilnya saja[16]. Sintak dan semantik PHP memiliki kesamaan dengan bahasa C, JAVA, PERL, dengan menambahkan tambahan *tag* `<?php` sebagai pembuka dan diakhiri dengan `?>`, pasangan kedua kode inilah yang berfungsi sebagai *tag* kode PHP. Berdasarkan *tag* inilah pihak server dapat memahami kode PHP dan kemudian memprosesnya, hasilnya dikirim ke browser[16].

PHP pada zaman sekarang kian makin berkembang karena maraknya penggunaan website dan perkembangannya. PHP banyak digunakan orang karena sifatnya yang *'user-friendly'* yaitu mudah dipahami karena kesederhanaanya dan disisi lain web berbasis PHP sangat aman dibandingkan dengan aplikasi dari pemrograman lain.

2.11 XAMPP

XAMPP adalah sebuah perangkat lunak web server Apache yang didalamnya sudah tersedia *database* server MySQL dan mendukung PHP programming. XAMPP merupakan singkatan dari X(untuk empat sistem operasi), Apache, MySQL, PHP, Perl[16].

2.12 Apache

Apache merupakan salah satu modul dalam perangkat lunak XAMPP yang berfungsi sebagai web server. Aplikasi ini tersedia untuk berbagai sistem operasi, termasuk Unix, GNU, FreeBSD, Linux, Solaris, Novell NetWare, AmigaOS, Mac[16].

2.13 MySQL

SQL ini merupakan bahasa standar yang digunakan untuk *Relational Database Management Systems* (RDBMS). Pernyataan-pernyataan SQL digunakan untuk melakukan beberapa tugas seperti update data atau menampilkan data dari basis data. Untuk dapat melakukan tugas tersebut diperlukan perangkat lunak *Database Management Systems* (DBMS) seperti MySQL, Oracle, Microsoft SQL Server, Microsoft Access, dan sebagainya. Setiap software *database* mempunyai bahasa perintah / sintaks yang berbeda, namun pada prinsipnya mempunyai arti dan fungsi yang sama[16]. MySQL memiliki keunggulan mulai dari aksesnya yang gratis, keamanan yang baik dan sangat mendukung transaksi.MySQL.

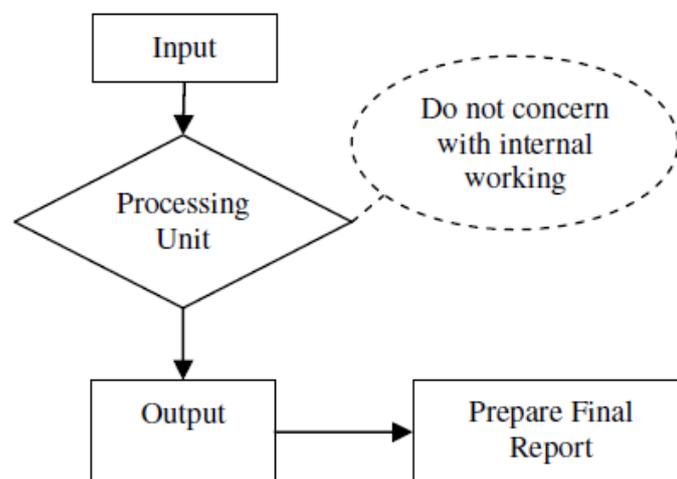
2.14 *Black Box Testing*

Black Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program[22]. Dalam *black box testing* cenderung untuk menemukan hal-hal berikut:

- a. Fungsi yang tidak benar atau tidak ada.

- b. Kesalahan antarmuka (*interface errors*).
- c. Kesalahan pada struktur data dan akses basis data.
- d. Kesalahan performa (*performance errors*).
- e. Kesalahan inisialisasi dan terminasi.

Berikut pada Gambar 2.14 gambaran dari *black box testing*, dimana *user* menyatakan masukan dan keluaran pada setiap unit proses dan didapatkan hasil dari setiap proses tersebut apakah sesuai dengan yang dinyatakan atau tidak.



Gambar 2.14 Gambaran Pengujian *Black Box* [22]

2.15 Confusion Matrix

Confusion matrix merupakan sebuah metode perhitungan yang digunakan untuk mencari keakuratan pada hasil klasifikasi[3]. Dalam Tabel 2.4 ditampilkan Tabel *confusion matrix*.

Tabel 2.4 Aturan Nilai dalam *Confusion Matrix*

Kelas		Prediksi	
		1	0
Kelas sebenarnya	1	TP	FN
	0	FP	TN

Keterangan:

- True Positive* (TP), merupakan jumlah dokumen dari kelas 1 yang benar diklasifikasikan sebagai kelas 1.
- False Positive* (FP), merupakan jumlah dokumen dari kelas 0 yang salah diklasifikasikan sebagai kelas 1.
- False Negative* (FN), merupakan jumlah dokumen dari kelas 1 yang salah diklasifikasikan sebagai kelas 0.
- True Negative* (TN), merupakan jumlah dokumen dari kelas 0 yang benar diklasifikasikan sebagai kelas 0.

Untuk menghitung akurasi digunakan perhitungan *Overall Accuracy* dengan membagi jumlah hasil klasifikasi yang bernilai benar (Jumlah dari diagonal di *confusion matrix*) dengan total dari seluruh baris dan kolom dari *confusion matrix*[25] atau dapat dilihat pada rumus 2.9.

$$accuracy = \frac{\text{correctly predicted class}}{\text{total testing class}} \times 100\% \quad (2.9)$$

Kemudian rumus untuk menghitung precision, recall, akurasi, dan error rate dapat dilihat dari rumus 2.10 sampai 2.12[26] :

- Precision*

Precision adalah nilai ketepatan dari nilai kelas hasil klasifikasi yang dilakukan sistem .

$$precision = \frac{TP}{TP + FP} \quad (2.10)$$

b. *Recall*

Recall adalah tingkat kemampuan sistem untuk dapat mengambil informasi.

$$recall = \frac{TP}{TP + FN} \quad (2.11)$$

c. *F1-Score/F1-Measure*

F1-Score adalah tingkat evaluasi bagi metode klasifikasi yang menggabungkan nilai dari *precision* dan *recall*.

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall} \quad (2.12)$$

Kemudian rumus untuk menghitung *precision*, *recall*, dan *f1-score* jika jumlah kelas lebih dari 1 adalah menggunakan rumus rata-ratanya. Dapat dilihat dari rumus 2.13 sampai 2.15 [26] :

a. *Precision*

$$precision = \frac{\sum_1^l \frac{TP}{TP + FP}}{l} \quad (2.13)$$

b. *Recall*

$$recall = \frac{\sum_1^l \frac{TP}{TP + FN}}{l} \quad (2.14)$$

c. *F1-Score/F1-Measure*

$$F1 - Score = \frac{\sum_1^l \frac{2 \times precision \times recall}{precision + recall}}{l} \quad (2.15)$$

Dengan keterangan tambahan :

l = Jumlah Kelas.

