

### 3. Perancangan Sistem

#### 3.1 Deskripsi Sistem

Pada bab ini akan dibahas mengenai analisa dan perancangan perangkat lunak untuk pembangunan daftar *stopword* secara otomatis yang akan digunakan untuk proses kategorisasi dokumen. Perancangan implementasinya berupa diagram aliran data (DAD). Diagram aliran data dipakai sebagai alat bantu pengembangan sistem yang memiliki level-level sesuai dengan tingkat kedalamannya. Diagram aliran data dipakai karena mempunyai sifat yang dapat menjamin kejelasan sistem yang digambarkan, kelengkapan penggambaran dan menghindari keambiguan.

#### 3.2 Analisa sistem

Pada tugas akhir ini akan dibangun sebuah perangkat lunak yang akan digunakan untuk pembangunan daftar *stopword* secara otomatis dan akan digunakan juga Weka sebagai *tool* Data Mining untuk proses pengkategorisasian dokumen, guna menguji tingkat keakuratan dari daftar *stopword* yang dibangun dengan melihat nilai F-measure.

Metode yang digunakan untuk pembangunan daftar *stopword* adalah *Term-Based Random Sampling*. Algoritma *Term-Based Random Sampling* ini didasarkan pada seberapa informatifnya suatu term dalam suatu dokumen. Dengan ini bisa ditentukan tingkat kepentingan dari term-term yang ada dalam dokumen, term dengan tingkat kepentingan yang kecil maka akan cenderung dikategorikan ke dalam daftar *stopword*. Untuk menghitung tingkat kepentingan dari term-term yang ada, maka digunakan *Kullback-Leibler divergence measure* untuk menghitung nilainya.

Pada aplikasi ini akan dilakukan proses stemming terlebih dahulu yaitu pengembalian kata-kata yang ada dalam dokumen kedalam bentuk dasarnya. Kemudian dilakukan pembangunan daftar *stopword* dengan melihat bobot masing-masing term dan juga memperhatikan nilai thresholdnya.

#### 3.3 Kebutuhan Fungsionalitas Sistem

- a. Penggunaan sistem menentukan direktori utama penyimpanan data mentah berupa file txt yang akan digunakan dalam proses pembangunan daftar *stopword*.
- b. Pengguna sistem menginputkan jumlah sampel/iterasi.
- c. Pengguna menentukan menentukan threshold untuk jumlah feature yang akan dipilih dalam penentuan daftar *stopword*.

### **3.4 Konfigurasi Sistem**

#### **3.4.1 Spesifikasi Perangkat Keras**

Berikut merupakan daftar spesifikasi perangkat keras yang digunakan dalam pembangunan sistem :

- a. Processor AMD Athlon(tm) 64 X2 Dual Core Processor 5200+
- b. Memori 2 GB RAM
- c. Harddisk Seagate 7200 RPM 80 GB
- d. Monitor LG 17"
- e. Keyboard dan Mouse

#### **3.4.2 Spesifikasi Perangkat Lunak**

Berikut perangkat lunak yang digunakan saat menjalankan sistem ini:

- a. Windows XP Service Pack 2
- b. XAMPP Version 1.5.1
- c. PHP Version 5.2.1
- d. MySQL versi 5.0.21
- e. Apache
- f. Dreamweaver MX 2004

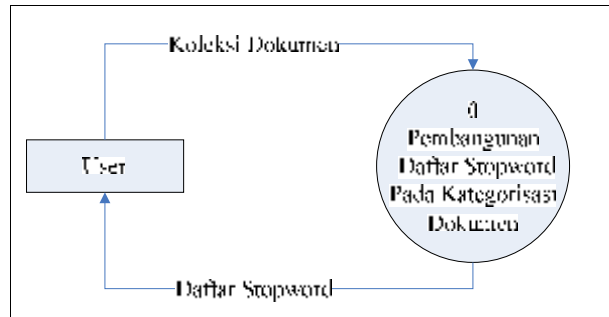
### **3.5 Perancangan Sistem**

#### **3.5.1 Diagram Aliran Data**

Pada bagian ini akan membahas mengenai perancangan sistem menggunakan Diagram Aliran Data (DAD) karena sifatnya yang menjamin kejelasan sistem yang digambarkan dan kelengkapan penggambaran. Diagram aliran data tersebut dapat terlihat pada level 0 yang menggambarkan sistem perangkat lunak secara global dan DAD level 1 sebagai pecahan dari DAD level 0 dan begitu seterusnya.

#### **3.5.2 Diagram Konteks**

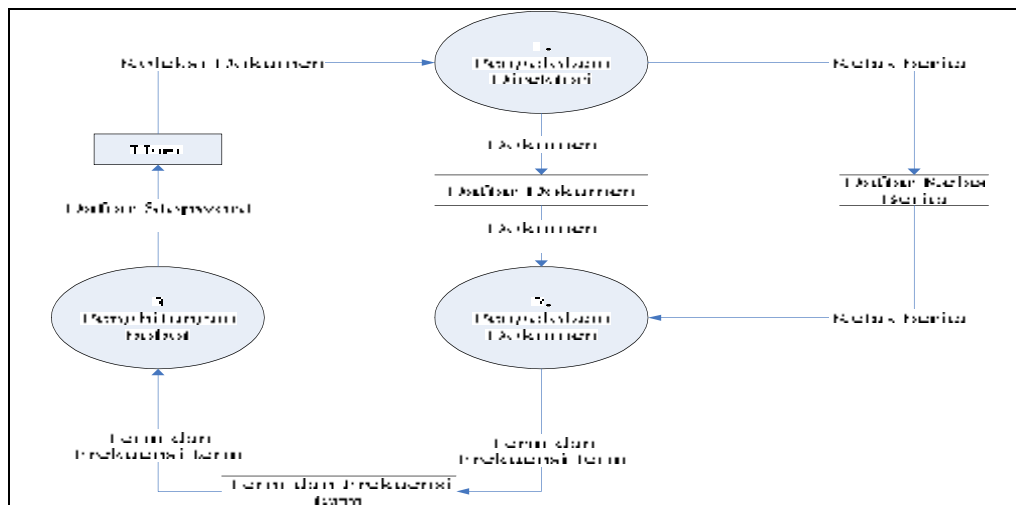
Diagram konteks dapat juga disebut dengan Diagram Aliran Data level 0. Diagram kontekas berisi penjelasan umum atau global tentang proses yang terjadi dalam sistem yang menggunakan interaksi antara sistem dan entity luar. Adapun diagram konteks dari sistem yang akan dibangun dapat dilihat pada gambar dibawah ini :



Gambar 3-1 Diagram Konteks

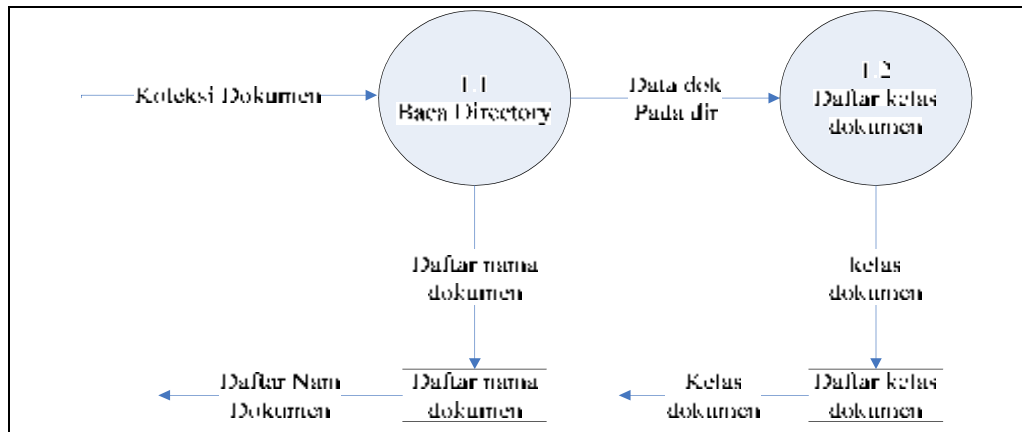
Diagram konteks diatas secara umum bisa dijelaskan bahwa sistem pembangunan daftar *stopword* pada kategorisasi dokumen berinteraksi dengan satu entitas luar yaitu user. User berperan untuk memberikan masukan kepada sistem berupa koleksi dokumen yang berformat txt, yang digunakan sebagai data untuk memperoleh daftar *stopword*.

### 3.5.3 DFD level 1



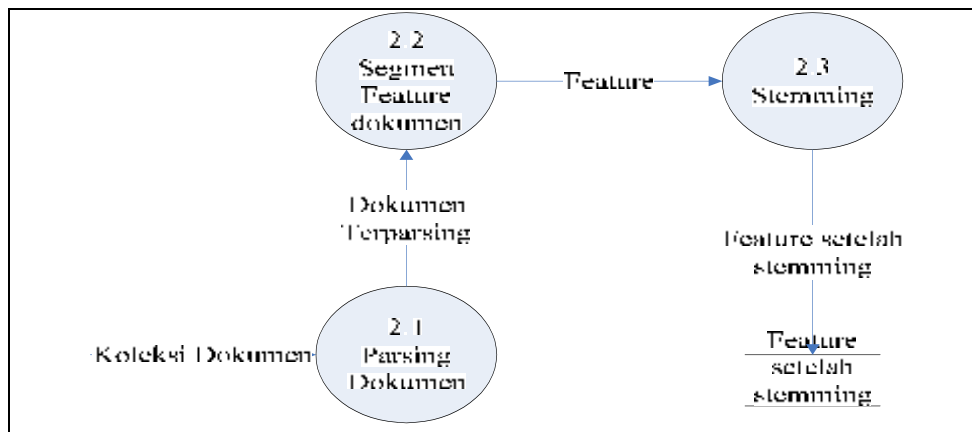
Gambar 3-2 DFD level 1

### 3.5.4 DFD level 2 proses 1



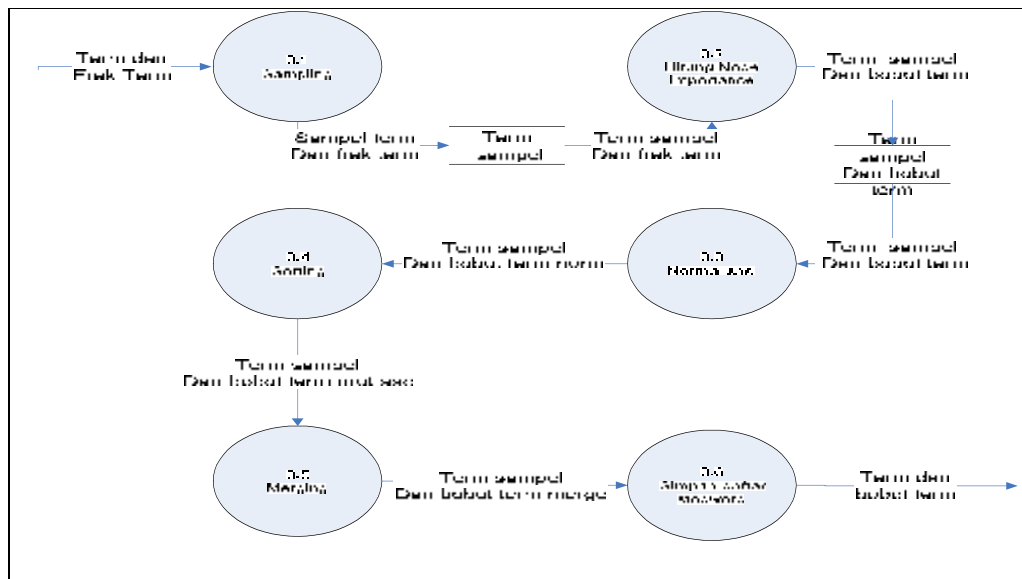
Gambar 3-3 DFD level 2 proses 1

### 3.5.5 DFD Level 2 proses 2



Gambar 3-4 DFD Level 2 proses 2

### 3.5.6 DFD Level 2 proses 3



Gambar 3-5 DFD Level 2 proses 3

### 3.5.5 Kamus Data

Kamus data merupakan suatu daftar yang mendefinisikan semua elemen data yang terlibat dalam sistem. Kamus data merupakan spesifikasi lebih rinci dari data yang terlibat dalam aliran informasi adapun kamus data yang digunakan pada sistem diatas adalah sebagai berikut :

Dokumen	=	Id_dok + Frasa + jml_frasa_kata + kategori
Nama file	=	{daftar_frasa_kata}
Kelas berita	=	Id_kelas + kelas
Kata/feature	=	{karakter}
Jml_kata	=	{number}
Id_dok	=	{number}
Id_kelas	=	{number}
Kelas	=	{karakter}
Karakter	=	{'0'..'9'   'A'..'Z'   'a'..'z'}
Number	=	{0..9}

### 3.5.6 Spesifikasi Proses

Spesifikasi proses proses digunakan untuk menjelaskan level terendah dari suatu proses yang ada pada diagram aliran data. Spesifikasi dari proses-proses yang terlibat dalam perancangan sistem ini adalah sebagai berikut :

*Tabel 3-1 Spesifikasi proses baca direktori*

No. Proses	1.1
Nama Proses	Baca direktori
Deskripsi	Proses di sini merupakan pembacaan tiap direktori tempat dokumen disimpan dan kemudian akan disimpan dalam storage device
Input	Dokumen pada direktori
Output	Daftar kategori dokumen
Logika proses	<u>For each (dir pada direktori)</u> <u>Open</u> dir ← (tiap direktori pada path yang telah ditentukan) While (List_dir ≠ Null) Readdir ← (nama direktori) Endwhile <u>Endfor</u> <b>Input</b> (nama berita)

*Tabel 3-2 Spesifikasi proses daftar Kelas Dokumen*

No. Proses	1.2
Nama Proses	Daftar kelas dokumen
Deskripsi	Proses untuk menyimpan kata sebagai label tiap berita
Input	Nama direktori
Output	Daftar kelas/kategori
Logika proses	<b>Input</b> (nama direktori sebagai kategori label dokumen) Read(nama dir) as kategori

*Tabel 3-3 Spesifikasi proses parsing*

No. proses	2.1
Nama proses	Parsing
Deskripsi	Pembacaan terhadap setiap kata-kata yang ada pada masing-masing dokumen
Input	Dokumen txt
Output	Dokumen terparsing
Logika proses	<u>Input</u> (dokumen) <u>While</u> (!Eof) Read(dokumen) parsing Input to Database <u>Endwhile</u>

*Tabel 3-4 Spesifikasi proses segmen feature dokumen*

No. proses	2.2
Nama proses	Segmen <i>Feature</i> Dokumen
Deskripsi	Melakukan pemisahan isi dokumen per <i>feature</i> atau kata

Input	terparsing
Output	<i>Feature</i> atau kata
Logika proses	Input(text) While(!Eof) If space found $\leftarrow$ explode(text) Input to database Endwhile

*Tabel 3-5 Spesifikasi proses stemming*

No. proses	2.3
Nama Proses	Stemming
Deskripsi	Pengembalian kata-kata kedalam bentuk dasarnya pada tiap teks
Input	<i>Feature</i> atau kata
Output	<i>Feature</i> dalam bentuk kata dasar
Logika proses	Input (term) While(!Eof) If term not (tidak bentik dasar) Stem_porteer(term) Endwhile

*Tabel 3-6 Spesifikasi proses Sampling*

No. proses	3.1
Nama Proses	Sampling
Deskripsi	Memilih sampel dokumen secara random
Input	<i>Feature</i> atau kata
Output	<i>Feature</i> yang termasuk sampel dokumen
Logika proses	for (y=1 to jumlah_sampel) w_random $\leftarrow$ random_choose (feature)  if (feature_in_doc = w_random) retrieve_all_in_doc_contain_w_random (feature) endif  endfor

*Tabel 3-7 Spesifikasi Hitung Node Importance*

No. proses	3.2
Nama Proses	Hitung Node Importance
Deskripsi	Proses penghitungan bobot tiap-tiap term dalam dokumen yang terambil untuk melihat tingkat kepentingan suatu term dalam sampel dokumen
Input	<i>term</i> dalam sampel dokumen atau kata
Output	<i>Term</i> dan bobot
Logika proses	input(term,frek)

	count $\leftarrow$
	$\omega(t) = P_x * \log_2 \frac{P_x}{P_c}$

*Tabel 3-8 Spesifikasi proses Normalisasi*

No. proses	3.3
Nama Proses	Normalisasi
Deskripsi	Membagi masing-masing bobot dengan bobot maksimum, sehingga bobot masing-masing term berada dalam rentang [1,0]
Input	<i>Term</i> dan bobot
Output	<i>Term</i> dan bobot ternormalisasi
Logika proses	Normalisasi (bobot term) For i = 1 to jumlah_term Bobot_norm $\leftarrow$ Bobot/bobot_max Endfor

*Tabel 3-9 Spesifikasi proses sorting*

No. proses	3.4
Nama Proses	Sorting
Deskripsi	Proses mengurutkan bobot term secara ascending
Input	<i>term</i> dan bobot
Output	<i>term</i> terurut ascending sesuai bobot
Logika proses	Ranking (bobot term)

*Tabel 3-10 Spesifikasi proses merging*

No. proses	3.5
Nama Proses	Merging
Deskripsi	Menggabungkan term-term yang sama, dengan menghitung bobot rata-rata dari term tersebut sebagai bobot term yang baru
Input	<i>term</i> dan bobot
Output	<i>term</i> dengan kondisi telah di merging
Logika proses	Read (term[i]) While (term[i] $\neq$ Null) Check(term[i] per dokumen) If (term[i] = term[i+1]) A $\leftarrow$ Merge(term[i], i+1) Endif Count(bobot(A))

*Tabel 3-11 Spesifikasi proses simpan daftar stopwords*

No. proses	5.6
Nama Proses	Simpan daftar stopwords



Deskripsi	Proses pengambilan daftar stopwords
Input	<i>Term</i> atau bobot
Output	<i>term</i> dan bobot
Logika proses	Read (term,bobot) If (id_urut > threshold) Simpan_to_database Endif