BAB III

METODE PENELITIAN

3.1 Pengumpulan Data

Penelitian ini pada dasarnya lebih mengedepankan aspek implementasi dan analisis suatu teori, tentang *stemming* dengan menggunakan pendekatan algoritma Nazief Adriani dan Porter. Seperti pada penelitian-penelitian tentang analisis atau pembuktian suatu teori yang banyak menerapkan metode studi kepustakaan, demikian hal nya pada penelitian ini, penulis lebih banyak melakukan studi kepustakaan dengan mengambil referensi dari internet, *ebook*, *paper*, jurnal, skripsi, tesis, maupun buku-buku yang berkaitan dengan algoritma *stemming*, khusunya pada bahasa Indonesia, pembentukan kata pada Bahasa Indonesia, dan berbagai literatur ilmiah lainnya.

Referensi utama dalam mengimplementasikan algoritma Nazief & Adriani adalah penelitian yang dilakukan oleh Jelita Asian B.Comp. Sc.(Hons.) (2007) dalam tesisnya yang berjudul "*Effective Techniques for Indonesian Text Retrieval*". Dalam penelitiannya tersebut, Asian meneliti tentang *information retrieval* pada bahasa Indonesia, dimana salah satu bahasannya adalah implementasi algoritma Nazief & Adriani.

Selain itu, referensi utama dalam mengimplementasikan algoritma Porter adalah penelitian yang dilakukan oleh Fadillah Z Tala (2003) dalam penelitiannya dengan judul "*A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*", yang meneliti tentang implementasi algoritma Porter untuk bahasa Indonesia.

Dalam penelitian ini, implementasi algoritma Nazief Adriani memerlukan kamus kata dasar yang diperoleh dari internet, dalam penelitian ini penulis menggunakan kamus kata dasar yang diunduh dari internet yang memiliki jumlah kata paling banyak yaitu 28.532 kata[1]. Kata dasar tersebut kemudian disimpan dalam database yang nantinya akan digunakan dalam proses *stemming*.

Sebelum proses *stemming* dijalankan, terlebih dahulu dilakukan penghapusan tanda baca. Berikut adalah daftar tanda baca yang digunakan:

Tabel. 3.1 Daftar Tanda Baca

No	Tanda Baca
1	.
2	,
3	;
4	:
5	-
6	–
7	…
8	(
9	)
10	?
11	!
12	(
13	)
14	[
15	]
16	“
17	‘
18	/
19	`

[1] http://sekedar-tutorial.blogspot.co.id/2012/11/program-stemming-bahasa-indonesia.html

No	Tanda Baca
20	0
21	1
22	2
23	3
24	4
25	5
26	6
27	7
28	8
29	9
30	_
31	%
32	~
33	@
34	#
35	$
36	^
37	&
38	*
39	+
40	=
41	<
42	>
43	[
44	]
45	{
46	}
47	|
48	\

Sumber: Data primer yang diolah

Jumlah karakter yang tidak terlalu banyak, dianggap lebih efisien jika disimpan dalam bentuk file teks (.txt). Tanda baca tersebut kemudian disimpan dalam sebuah file teks (.txt), yang nantinya akan dibaca oleh sistem, kemudian dijadikan sebagai pembanding untuk mengetahui adanya tanda baca dalam sebuah teks, kemudian menghapusnya.

Proses selanjutnya adalah penghapusan *stopword list*. *Stopword list* merupakan kata umum yang dianggap tidak memberikan informasi penting, sehingga keberadaannya bisa diabaikan, misalnya kata : yang, dan, yaitu, adalah dan lain sebagainya. Dalam penelitian ini penulis menggunakan *stopword list* sebagaimana yang digunakan pada penelitian Tala (2003) yang terdiri dari 758 kata[2]. Seperti halnya tanda baca, *stopword list* tersebut disimpan dalam bentuk file teks yang nantinya dibaca oleh sistem. Dengan membandingkan kata dalam *stopword list* dengan kata dalam korpus, jika ditemukan, kata tersebut akan dihapus.

Dalam penelitian ini digunakan 26 dokumen uji yang diperoleh dari situs berita online (detik.com) pada tanggal 27 September 2015. Dokumen tersebut diunduh menggunakan *web crawler*, yang dibuat menggunakan *library* php bernama *PHP simple HTML DOM parser*. Dokumen yang diunduh akan di simpan disimpan dalam bentuk file XML (.xml), dengan struktur sebagai berikut:

```
<document>
    <title>Judul Berita</title>
    <content>Isi Berita</content>
</document>
```

[2] Fadillah Z Tala, *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Institute for Logic, Language and Computation Universeit Van Amsterdam, 2003 hal. 39

Untuk memudahkan pemanggilan dokumen ketika proses stemming, indeks dokumen tersebut disimpan dalam database.

Konten atau isi dari dokumen tersebut yang akan di *stemming* menggunakan algoritma Nazief Adriani dan Porter, untuk kemudian hasilnya dilakukan analisis atas hasil yang diperoleh.

3.2 Analisis dan Perancangan Sistem

Sebuah sistem dibangun untuk mengimplementasikan proses pencarian akar kata (stemming) yang merupakan sub proses dari sistem temu kembali informasi dengan menerapkan algoritma Porter dan Nazief & Adriani. Aplikasi ini dibangun berbasis web dengan menggunakan bahasa pemrograman PHP dan database MySQL.

Setiap dokumen uji yang diunduh dari situs berita *online* (detik.com) memiliki identitas berupa id yang digunakan sebagai *key* untuk memanggil dokumen uji tertentu kemudian menampilkan isinya. Langkah pertama yang dilakukan adalah mengubah semua huruf kapital menjadi huruf kecil (*lower case*), dilanjutakan dengan menghilangkan semua tanda baca dan angka, sehingga didapat serangkaian kata yang membentuk satu kalimat panjang. Dari serangkaian kata tersebut, diubah menjadi sekumpulan kata dengan memisahkannya berdasarkan tanda spasi. Langkah berikutnya adalah mengalukan pengecekan apakah ada diantara kata-kata tersebut yang termasuk dalam *stopword list*. Jika ditemukan, kata tersebut akan dihilangkan, sehingga didapat sekumpulan kata baru. Demikin juga untuk kata-kata yang muncul lebih dari satu kali (*double words*), akan dieliniasi dan hanya satu yang akan di proses.

Selanjutnya, kata-kata tersebut akan dilakukan *stemming* menggunakan *stemmer* yang Nazief & Adriani dan Porter, sehingga akan didapatkan akar kata (*root word*). Tidak semua hasil yang didapatkan merupakan akar kata (*root word*) yang benar. Beberapa kesalahan *stemming* yang mungkin terjadi, yaitu proses penghilangan imbuhan kurang (*understemming*), imbuhan yang dihilangkan berlebihan (*overstemming*) dan kata tidak berubah atau tidak ada imbuhan yang di buang (*unchange*). Hasil tersebut akan disimpan dalam database untuk memudahkan proses analisis. Demikin juga waktu proses akan dicatat untuk mengukur tingkat efisiensi proses *stemming*.

3.2.1 Kebutuhan Perangkat Lunak

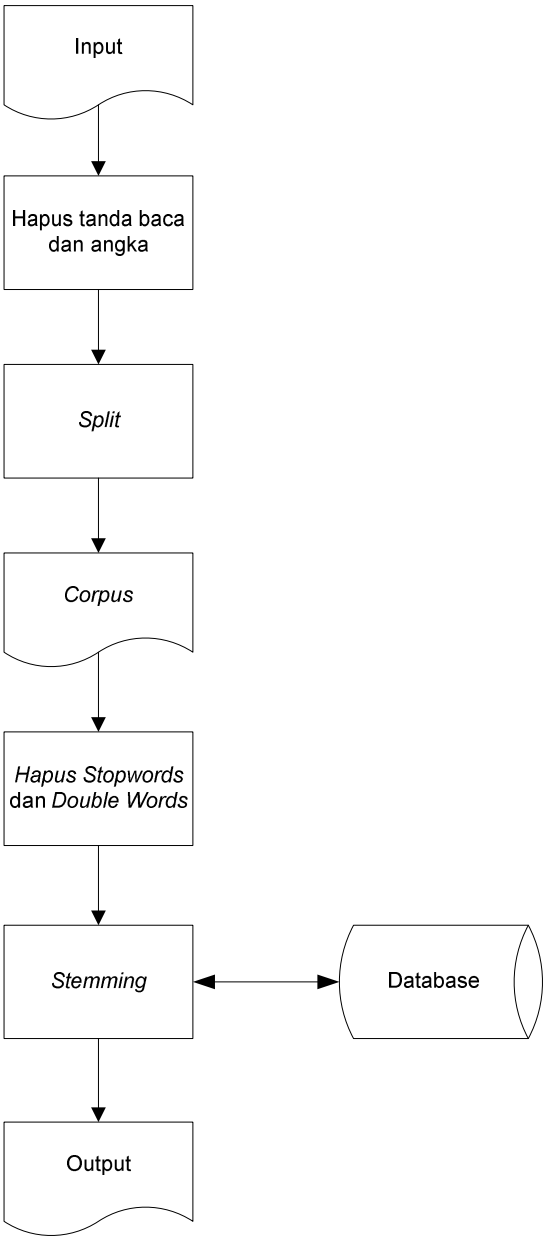
Aplikasi ini berbasis web, dibangun menggunakan perangkat-perangkat lunak, diantaranya:

1. Bahasa pemrograman PHP version 5.6.7
2. Database MySQL 5.0.11
3. *Web Server* Apache 2.2.29
4. *Web browser* yang mendukung HTML 5, CSS3, dan Jquery, dalam hal ini penulis menggunakan Safari 8.0
5. *Text editor*, penulis menggunakan Brackets.

3.2.2 Flowchart

Pengembangan aplikasi ini dimaksudkan untuk memudahkan dalam mengimplementasikan dan menganalisis algoritma *stemming* yang diujikan dalam

penelitian ini, yaitu Nazief & Adriani dan Porter. Berikut adalah *flow chart* implementasi kedua algoritma *stemming* tersebut:



Gambar 3.1 *Flow Chart* implementasi algoritma *stemming*

Seluruh tanda baca dan angka pada dokumen input dihapus, dengan mengacu pada Tabel 3.1. Selanjutnya akan di *split* berdasarkan tanda spasi sehingga menjadi sebuah kumpulan kata (*corpus*). *Corpus* ini akan diperiksa ada tidaknya

kata yang masuk daftar *stopwords* dengan mengacu data yang ada didatabase, jika ditemukan kata akan dihapus atau tidak akan di proses. Demikian juga untuk kata yang muncul lebih dari satu kali (*double words*), hanya satu kata yang akan diproses dan yang lainnya akan dieliminir sehingga membentuk *corpus* baru. *Corpus* inilah yang kemudian akan dilakukan proses *stemming* yang menghasilkan akar kata. Hasil *stemming* akan disimpan dalam database untuk memudahkan dalam proses analisis.

3.2.3 Rancangan Aplikasi

Untuk memudahkan implementasi dan analisis algoritma yang diujikan pada penelitin ini, maka dibangun sebuah aplikasi berbasis web yang terdiri atas 6 halaman, yaitu Dokumen, Stemming, Crawler, Hasil, Kata Dasar dan Stopword.

Halaman Dokumen berisi daftar dokumen yang diujikan pada penelitian ini. Setelah isi ditampilkan, ada 2 pilihan *stemmer* yang digunakan pada penelitian ini.

Halaman Stemming berisi *form* untuk melakukan uji *stemming* dengan memberikan input secara manual, kemudian akan di uji menggunakan *stemmer* Nazief & Adriani dan Porter.

Halaman Crawler menampilkan *index* berita terbaru pada situs berita detik.com. Terdapat tombol Unduh yang akan mengunduh semua berita yang terindeks beserta kontennya kemudian akan menyimpannya dalam database ketika tombol ini di klik. Konten dari berita tersebut juga akan disimpan dalam bentuk file XML (.xml).

Halaman Hasil menampilkan hasil dari proses *stemming* terhadap dokumen-dokumen yang diujikan yang disajikan dalam bentuk tabel,

menampilkan indeks, judul dokumen, jumlah seluruh kata, jumlah kata ganda (*double words*), *stopwords*, jumlah kata yang di proses (*stemming words*), jumlah kata yang benar, kata yang mengalami kesalahan stemming (*understemming*, *overstemming*, dan *unchange*). Sehingga dapat dihitung prosentase kesalahannya. Dengan pendekatan bahwa hasil *stemming* yang benar akan meningkatkan nilai *recall*, yang tentu akan memberikan hasil sistem temu kembali informasi yang lebih relevan. Kecepatan proses *stemming* menunjukkan efisiensi waktu yang diperlukan.

Halaman Kata dasar berisi daftar kata dasar yang digunakan sebagai data pembanding pada proses *stemming*.

Halaman Stopwords berisi daftar kata yang termasuk dalam *stopword list*. Kata-kata pada daftar ini dianggap sebagai kata yang tidak akan berpengaruh jika kata-kata tersebut dihilangkan. Kata-kata yang terdeteksi sebagai *stopword* pada proses *stemming*, akan dihilangkan.

3.3 Analisis Hasil

Proses *stemming* yang baik diharapkan akan menghasilkan akar kata yang benar. Namun demikian, beberapa faktor mempengaruhi kesalahan proses stemming ini, diantaranya *overstemming*, *understemming*, maupun *unchange*.

Hasil yang diperoleh dari proses *stemming* akan diperiksa secara manual untuk mengetahui jumlah kata yang mengalami kesalahan *stemming*, dihitung prosentase kesalahannya. Begitu juga akan dicatat waktu proses *stemming* tersebut.

Kesimpulan akhir akan diukur tingkat akurasinya menggunakan persamaan:

Akurasi = (RW / W) * 100%

dimana W adalah jumlah kata yang ter-*stem* dan RW adalah jumlah kata yang di-*stemming* dengan benar. Akurasi dinyatakan dalam persen (%).