

BAB II

DASAR TEORI

Tinjauan Pustaka yang akan dilakukan mencakup beberapa materi yaitu tentang *Social Network*, *Social Network Analysis (SNA)*, Algoritma Nazief dan Adriani, *Website*, Pemrograman Berorientasi Objek, *PHP*, *MySQL*, *UML*, Skala Likert, Pengujian *Black Box* dan Pengujian *Beta*. Berikut adalah penjelasan singkatnya:

2.1 *Social Network*

Social Network biasanya didefinisikan sebagai grafik yang mewakili hubungan sosial antara individu atau organisasi [3]. Setiap *node* dalam grafik mewakili individu atau organisasi dan biasanya juga disebut *actor* atau *vertex*. Dan yang menghubungkan dua *node* biasanya disebut *edge* atau *tie*. Menggunakan grafik untuk mewakili data sosial memungkinkan seorang analis sosial untuk secara lengkap dan cermat dapat mendeskripsikan, memanipulasi, dan menganalisis informasi struktural yang disematkan dalam hubungan sosial. Selain itu, 3 konsep teori grafik memberi peneliti sebuah kerangka kerja matematis dan sistematis yang dapat memperluas metodologi para peneliti ke bidang lain. Secara umum, jaringan sosial dapat digunakan untuk mewakili, mengidentifikasi, dan mengukur jenis korelasi antara entitas apa pun, seperti kata, halaman web, orang, organisasi, hewan, sel, komputer, dan informasi lainnya atau entitas pemrosesan pengetahuan [8]. Dengan demikian, *Social Network* memiliki aplikasi yang luas dan sukses dalam sosiologi, epidemiologi, biologi, kriminologi, dan ekonomi [7].

Saat ini, ada dua jenis pendekatan pengumpulan data *Social Network* yaitu *elicitation* dan *registration* [18]. *Elicitation* adalah proses mendapatkan data *Social Network* dengan cara kuesioner atau survei. Sedangkan *registration* adalah proses mendapatkan data *Social Network* melalui penggalian dari informasi yang terdaftar.

2.2 *Social Network Analysis (SNA)*

Ada beberapa pendapat dari para peneliti tentang pengertian *SNA*. Krebs mendefinisikan bahwa “*SNA* adalah proses pemetaan dan pengukuran relasi antara orang ke orang” [8], Selanjutnya Freeman mendefinisikan “*SNA* sebagai teknik yang fokus mempelajari pola interaksi pada manusia yang tidak terlihat secara

eksplisit” [5]. Dan kemudian Scott mendefinisikan “SNA sebagai sekumpulan metode untuk menginvestigasi aspek relasi pada struktur sosial” [16]. Berdasarkan ketiga definisi tersebut, secara garis besar memiliki kesamaan makna, yaitu mengarah pada proses analisis jaringan sosial yang berkaitan dengan bentuk struktur dan pola interaksi entitas di dalamnya. Dari pengertian tersebut dapat dikatakan bahwa SNA lebih menekankan pada interaksi antar entitas di dalamnya daripada entitas itu sendiri, dengan kata lain SNA lebih banyak membahas hubungan antar aktor daripada atribut aktor tersebut [16]. Pola interaksi antar entitas akan memberikan informasi baru. Namun bukan berarti entitas tidak ada gunanya sama sekali. Atribut pada entitas yang menjadi *node* pada graf memiliki informasi yang dapat membantu untuk membuat hipotesa atas fenomena yang terjadi.

2.3 Algoritma Nazief dan Adriani

Algoritma Nazief & Adriani dikembangkan pertama kali oleh Bobby Nazief dan Mirna Adriani. Algoritma ini berdasarkan pada aturan morfologi bahasa Indonesia yang luas, yang dikumpulkan menjadi satu grup dan di-enkapsulasi pada imbuhan/*affixes* yang diperbolehkan (*allowed affixes*) dan imbuhan/*affixes* yang tidak diperbolehkan (*disallowed affixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung *recording*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih. [13]

Algoritma ini mempunyai aturan imbuhan sendiri dengan model sebagai berikut [1]:

$$[[[AW +]AW +]AW +] \text{ Kata-Dasar } [[+AK][+KK][+P]$$

AW : Awalan

AK : Akhiran

KK : Kata ganti kepunyaan

P : Partikel

Tanda kurung besar menandakan bahwa imbuhan adalah opsional.

Dalam algoritma nazief dan adriani ada beberapa kombinasi awalan dan akhiran yang tidak diperbolehkan, yaitu kombinasi awalan dan akhiran yang ada dalam tabel 2.1. Namun ada satu pengecualian pada kombinasi prefiks “ke-“ dan surfiks “-i” yang boleh diterapkan pada kata “tahu” menjadi kata “ketahui”.

Tabel 2.1 Kombinasi Prefiks dan Surfiks yang tidak diperbolehkan

Awalan (Prefiks)	Akhiran (Surfiks)
be-	-i
di-	-an
ke-	-i -kan
me-	-an
se-	-i -kan
te-	-an

2.3.1 Aturan Peluluhan Kata Dasar

Ada beberapa kata dasar yang apabila dilekati oleh awalan “me(N)-“, “pe(N)-“, “pe(R)-“, “te(R)-“, “be(R)-“ akan mengalami peluruhan atau perubahan pada karakter awal dari kata dasar tersebut [9]. Sebagai contoh kata “tanya”, karakter awal dari kata “tanya” akan berubah apabila ditambahkan awalan “me-“ dan menjadi “menanya”. Begitu juga untuk beberapa kata dasar lainnya. Untuk melakukan proses *stemming* pada kata-kata tersebut harus mengikuti aturan peluruhan yang telah ditetapkan oleh algoritma [1]. Aturan-aturan tersebut dijelaskan pada tabel 2.2.

Tabel 2.2 Aturan Peluruhan Kata Dasar [1]

Aturan	Bentuk Awal	Peluruhan
1	berV...	Ber-V... be_rV...
2	berCAP... .	ber-CAP... .where C!=‘r’ and P!=‘er’
3	berCAerV... .	ber-CAerV... .where C!=‘r’
4	belajar...	Bel-ajar
5	beC1erC2...	Be-C1erC2...dimana C1!=‘r’ ‘l’}
6	terV...	Ter-V... te-rV...
7	terCP... .	ter-CP... .where C!=‘r’ and P!=‘er’
8	terCer...	Ter-Cer...dimana C!=‘r’
9	teC1erC2...	Te-C1erC2...dimana C1!=‘r’
10	me{1r w y}V...	Me-{1r w y}V...
11	mem{b f v}...	Mem-{b f v}...
12	mempe...	Mem-pe...
13	mem{rV V}...	Me-m{rV V}... me-p{rV V}...
14	men{c d j z}...	Men-{c d j z}...
15	menV...	Me-nV... me-tV...
16	meng{g h q k}...	Meng-{g h q k}...
17	mengV...	Meng-V... meng-V... meng-kV...
18	menyV...	Me-ny... meny-sV...
19	mempV...	Mem-pV...
20	pe{w y}V...	Pe-{w y}V...

21	perV...	Per-V... pe-rV...
22	perCAP. . .	per-CAP. . .where C!=‘r’ and P!=‘er’
23	perCAerV. . .	per-CAerV. . .where C!=‘r’
24	pem{b f v}...	Pem-{b f v}...
25	pem{rV V}...	Pe-m{rV V}... pe-p{rV V}
26	pen{c d j z}...	Pen-{c d j z}...
27	penV...	Pe-nV... pe-tV...
28	peng{g h q}...	Peng-{g h q}...
29	pengV...	Peng-V peng-kV
30	penyV...	Pe-nya peny-sV
31	pelV...	Pe-lV...;kecuali untuk kata “pelajar” menjadi “ajar”
32	peCP...	Pe-CP...dimana C!={r w y l m n} dan P!=‘er’
33	perCerV	Per-CerV... C!={r w y l m n}

Pada tabel 2.2 dapat dilihat aturan-aturan peluruhan kata dasar yang apabila dilekati oleh awalan “me-“, ”be-“, ”te-“, ”pe-“. Dimana pada kolom kedua dari tabel tersebut menjelaskan bentuk-bentuk kata dasar yang dilekati awalan “me-“, ”be-“, ”te-“, ”pe-“, sedangkan pada kolom ketiga menjelaskan perubahan-perubahan karakter pada kata dasar yang mungkin terjadi apabila algoritma telah menghilangkan awalan yang telah melekat pada kata dasar tersebut. Huruf “V” pada tabel tersebut menunjukkan huruf hidup atau huruf vocal, huruf “C” menunjukkan huruf mati atau konsonan, huruf “A” menunjukkan huruf vocal atau huruf konsonan dan huruf “P” menunjukkan pecahan “er”. Sebagai contoh, jika algoritma menerima kata “menyusun”, maka proses *stemming* pada kata tersebut mengikuti aturan ke-16 pada tabel 2.2 yaitu “menyV...” dan perubahan menjadi “me-ny” atau “meny-sV...”. Berdasarkan aturan tersebut maka algoritma akan menghilangkan awalan “me-“ maka akan didapatkan kata “nyusun”, selanjutnya kata “nyusun” akan diperiksa ke dalam *database* kata dasar, karena kata “nyusun” bukan kata kata dasar maka tahap selanjutnya algoritma akan menghilangkan kata “meny-“ dan kemudian algoritma akan menambahkan huruf “s” di depan huruf “u”, maka akan didapatkan kata “susun”, selanjutnya kata “susun” akan diperiksa kedalam *database* kata dasar. Karena kata “susun” merupakan kata dasar maka kata tersebut akan diidentifikasi sebagai kata dasar. [11]

2.4 Website

Website adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman *web* yang saling berhubungan. Hubungan antara satu halaman *web* dengan halaman *web* yang lainnya disebut dengan *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext*.

Seiringan dengan perkembangan teknologi informasi yang begitu cepat, *website* juga mengalami perkembangan yang sangat berarti. Dalam pengelompokan jenis *web*, lebih diarahkan berdasarkan fungsi, sifat dan bahasa pemrograman yang digunakan. Jenis-jenis *web* berdasarkan sifatnya adalah:

1. *Website Dinamis*
Merupakan sebuah *website* yang menyediakan *content* atau isi selalu berubah-ubah setiap saat, misalnya *website* berita.
2. *Website Statis*
Merupakan *website* yang *contentnya* sangat jarang diubah, misalnya *web* profil organisasi.

Ditinjau dari segi bahasa pemrograman yang digunakan, *website* terbagi atas:

1. *Server Side*
Merupakan *website* yang menggunakan bahasa pemrograman yang tergantung kepada tersedianya *server*. Seperti PHP, ASP dan lain sebagainya. Jika tidak ada *server*, *website* yang dibangun menggunakan bahasa pemrograman di atas tidak dapat berfungsi sebagaimana mestinya.
2. *Client Side*
Website yang tidak membutuhkan *server* dalam menjalankannya, cukup diakses melalui browser saja, misalnya html [19].

2.5 Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek (PBO) adalah paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Model data berorientasi objek

dikatakan lebih mudah dipelajari bagi pemula. Selain itu PBO juga mudah dikembangkan dan mudah dirawat.

Selain itu, secara spesifik pengertian berorientasi objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemrograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu [15]:

1. Identifikasi
2. Klasifikasi
3. *Polymorphism* (polimorfisme)
4. *Inheritance* (pewarisan)

2.6 *Hypertext Preprocessor (PHP)*

Hypertext Preprocessor (PHP) adalah bahasa *server-side –scripting* yang menyatu dengan *HyperText Mark up Language (HTML)* untuk membuat halaman web yang dinamis. Karena *PHP* merupakan *server-side-scripting* maka sintaks dan perintah-perintah *PHP* akan dieksekusi di *server* kemudian hasilnya akan dikirimkan ke *browser* dengan format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. *PHP* dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web [2]

2.7 *MySQL*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS* yang *multithread* dan *multi-user*. *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia, dimana perusahaan tersebut memegang hak cipta atas hampir semua kode sumbernya [2]

2.8 *Unified Modeling Language (UML)*

UML (Unified Modeling Language) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak berorientasi objek. *UML* memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema basis data, dan komponen-komponen yang diperlukan dalam sistem perangkat lunak. *UML* juga merupakan sebuah bahasa standar untuk pengembangan sebuah perangkat lunak yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan perangkat lunak. *UML* terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram tersebut menggambarkan permasalahan maupun solusi dari suatu model. *UML* mempunyai 9 diagram, yaitu; *use-case, class, object, state, sequence, collaboration, activity, component, dan deployment diagram* [4]

2.9 **Skala Likert**

Skala Likert pertama kali dikembangkan oleh Rensis Linkert pada tahun 1932 dalam mengukur sikap masyarakat. Skala Likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok tentang kejadian atau gejala sosial.

Dengan menggunakan skala likert, maka variabel yang akan diukur dijabarkan menjadi dimensi, dimensi dijabarkan menjadi sub variabel kemudian sub variabel dijabarkan lagi menjadi indikator-indikator yang dapat diukur. Pernyataan atau pertanyaan dalam skala likert dapat berupa pernyataan positif maupun negatif [14].

Prosedur dalam membuat skala Likert adalah sebagai berikut:

1. Pengumpulan item-item yang cukup banyak dan relevan dengan masalah yang sedang diteliti, berupa item yang cukup terang disukai dan yang cukup terang tidak disukai.

2. Item-item tersebut dicoba kepada sekelompok responden yang cukup representatif dari populasi yang ingin diteliti.
3. Pengumpulan responsi dari responden untuk kemudian diberikan skor, untuk jawaban yang memberikan indikasi menyenangkan diberi skor tertinggi.
4. Total skor dari masing-masing individu adalah penjumlahan dari skor masing-masing item dari individu tersebut.
5. Responsi dianalisis untuk mengetahui item-item mana yang sangat nyata batasan antara skor tinggi dan rendah dalam skala total.

Untuk mempertahankan konsistensi internal dari pertanyaan maka item yang tidak menunjukkan korelasi dengan total skor tidak menunjukkan beda [14].

Tabel 2.3 Skor Jawaban

No	Jawaban	Skor
1	Sangat setuju	5
2	Setuju	4
3	Ragu-ragu	3
4	Tidak setuju	4
5	Sangat Tidak Setuju	5

2.10 Pengujian Sistem

Pengujian sistem adalah proses dimana dilakukannya percobaan sistem dengan tujuan untuk memastikan fungsi sistem berjalan dengan baik. Pada pengujian sistem akan menggunakan *Black Box Testing*, *Alpha Testing* dan *Beta Testing* untuk menguji sistem.

2.10.1 *Black Box Testing*

Metode ujicoba *black box* memfokuskan pada keperluan fungsional dari *software*. Ujicoba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya [12]

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

2.10.2 *Beta Testing*

Beta Testing atau pengujian beta dilakukan pada satu atau lebih pengguna akhir perangkat lunak dalam lingkungan yg sebenarnya, pengembang biasanya tidak ada pada pengujian ini. Pelanggan merekam semua masalah (*real* atau imajiner) yg ditemui selama pengujian dan melaporkan pada pengembang dalam rentang waktu tertentu.

Ada beberapa faktor yang diperhatikan dalam pengujian beta ini, diantaranya adalah jumlah peserta uji, durasi uji dan cakupan demografi. Jumlah peserta uji biasanya ditentukan oleh pengembang perangkat lunak sesuai kebutuhan dan langsung diambil dari target pengguna perangkat lunak yang dibuat. Dan diharapkan nantinya akan memberikan masukan sesuai kebutuhan pengguna perangkat lunak tersebut. [15]