BAB 2
LANDASAN TEORI

2.1 Dokumen Karya Tulis Ilmiah

Dokumen karya tulis ilmiah merupakan data, catatan yang tertulis atau tercetak dan mengandung keterangan atau informasi yang ditulis oleh seorang ilmuwan berupa hasil penelitian atau pengembangan sebuah ilmu pengetahuan, teknologi dan seni yang diperoleh dari kumpulan pengalaman dan pengetahuan sebelumnya [8]. Dokumen karya tulis ilmiah yang digunakan pada penelitian ini adalah dokumen karya tulis ilmiah skripsi Program Studi Teknik Informatika Universitas Komputer Indonesia.

Skripsi adalah karya tulis ilmiah yang ditulis oleh mahasiswa/i sebagai persyaratan akhir pendidikan akademis pada jenjang perguruan tinggi. Pada dokumen skripsi biasanya tidak disertakan dokumen khusus untuk abstrak, sedangkan pada dokumen skripsi yang telah melalui proses dokumentasi, biasanya disertakan dengan dokumen abstrak.

Penelitian ini menggunakan dokumen karya tulis ilmiah skripsi sebagai *dataset* yang dibuat kedalam beberapa kelas pada lembar sampul dan abstrak. Pada lembar sampul terdiri dari kelas seperti Judul Penelitian, Jenis Penelitian, Kalimat Pengajuan, Penulis (sampul), NIM (sampul), Program Studi, Fakultas, Universitas dan Tahun. Sedangkan untuk kelas pada lembar abstrak seperti Judul Halaman (abstrak), Judul Penelitian (abstrak), *Other*, Penulis (abstrak), NIM (abstrak), Isi Abstrak, Kata kunci. Berikut contoh kelas yang ada pada dokumen karya tulis ilmiah pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1 Kelas pada Lembar Sampul Skripsi

Lembar Sampul Skripsi	No	Kelas	Kelas
ANALISIS GRAY LEVEL DIFFERENCE METHOD DAN METODE NAÏVE BAYES MENGIDENTIFIKASI PENYAKIT LIDAH MANUSIA (1) **SKRIPSI** (2) Diajukan untuk Menempuh Ujian Akhir Sarjana (3) **RIEKAL FAHMI** (4) **10110482** (5) **PROGRAM STUDI TEKNIK INFORMATIKA** (6) **FAKULTAS TEKNIK DAN ILMU KOMPUTER** (7) **UNIVERSITAS KOMPUTER INDONESIA** (8) **2015** (9)	1	Judul Penelitian (sampul)	0
	2	Jenis Penelitian	1
	3	Kalimat Pengajuan	2
	4	Penulis (sampul)	3
	5	NIM (sampul)	4
	6	Program Studi	5
	7	Fakultas	6
	8	Universitas	7
	9	Tahun	8

Tabel 2.2 Kelas pada Lembar Abstrak Skripsi

Lembar Abstrak Skripsi	No	Nama Kelas	Kelas
ABSTRAK (9) **ANALISIS GRAY LEVEL DIFFERENCE METHOD DAN METODE NAÏVE BAYES MENGIDENTIFIKASI PENYAKIT LIDAH MANUSIA** (10) Oleh: (11) **Riekal Fahmi** (12) **10110482** (13) Lidah adalah kumpulan otot rangka pada bagian lantai mulut yang dapat membantu pencernaan makanan dengan mengunyah dan menelan. Lidah dikenal sebagai indera pengecap yang banyak memiliki struktur tunas pengecap. Jika lidah ini tak berfungsi, dengan sendirinya akan berpengaruh terhadap rasa makanan ataupun selera makan kita. Tidak berfungsinya lidah sebagaimana mestinya disebabkan adanya kelainan atau penyakit. Salah satu cara agar dapat membedakan ciri tersebut ialah dengan cara mengenali perbedaan tekstur pada citra Terdapat beberapa metode untuk memperoleh ciriciri tekstur dalam suatu citra, Salah satu metode untuk memperoleh ciri-ciri citra tekstur adalah Gray level difference atau bisa di singkat (GLDM). Ciri-ciri tekstur yang didapat dari metode ini diantaranya adalah kontras, Angular Singular Moment, energi, invers different moment dan Mean. Dari hasil ciri-ciri tersebut kemudian digunakan untuk klasifikasi dengan menggunakan Naïve Bayes yang menentukan hasil klasifikasi berdasarkan nilai probabilitas terbesar. Objek yang diuji adalah citra jenis Penyakit lidah manusia. Dari penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut : naïve bayes dapat melakukan klasifikasi citra berdasarkan tekstur yang diekstraksi dengan metode matriks gldm. Dikarenakan data hasil ekstraksi ciri matriks gldm adalah berupa data continue, atau biasa disebut data nominal, sehingga saat proses klasifikasi data hasil ekstraksi ciri tersebut dapat langsung digunakan sebagai inputan dalam klasifikasi naïve bayes. (14) Berdasarkan hasil pengujian, kesimpulan yang didapatkan adalah naïve bayes dapat mengklasifikasi citra dengan baik, dikarenakan data hasil ekstraksi ciri tekstur penyakit dengan metode Gray level difference moment memiliki interval jarak yang berjauhan antar kelasnya. Sehingga klasifikasi naïve bayes dapat berjalan dengan baik saat melakukan klasifikasi 85%. Kata kunci : tekstur citra, ekstraksi ciri, gldm matriks, klasifikasi naïve bayes (15)	1	Judul Halaman Abstrak	9
	2	Judul Penelitian (Abstrak)	10
	3	*Other*	11
	4	Penulis (Abstrak)	12
	5	NIM (Abstrak)	13
	6	Isi Abstrak	14
	7	Kata Kunci	15

2.2 Algoritma

Algoritma adalah runtunan langkah-langkah untuk menyelesaikan suatu masalah secara sistematis. Dan biasanya ditulis kedalam bentuk notasi-notasi yang

dapat diterjemahkan kedalam berbagai bahasa pemrograman. Notasi algoritmik dapat ditulis dengan notasi sebagai berikut:

1. Deskriptif

Notasi deskriptif biasanya digunakan oleh orang yang terbilang masih awam, dengan menggunakan kalimat yang dapat dijabarkan secara gampang. Tetapi sulit untuk diterjemahkan kedalam bahasa pemrograman.

2. *Pseudocode*

Pseudocode merupakan tiruan dari kode program yang menggambakan logika-logika sehingga dapat diterjemahkan kedalam bahasa pemrograman menjadi lebih mudah.

3. *Flowchart*

Flowchart merupakan bentuk algoritma yang digambarkan dalam bentuk alur dan menggunakan beberapa bentuk geometri untuk menggambarkan proses yang terjadi, seperti persegi panjang yang menyatakan proses, bentuk diamond yang menyatakan percabangan dan sebegainya. Dan penggunaan notasi ini tidak disarankan untuk kasus dengan skala yang besar, dan cenderung sulit untuk diterjemahkan kedalam bahasa pemrograman.

2.3 Ekstraksi Informasi

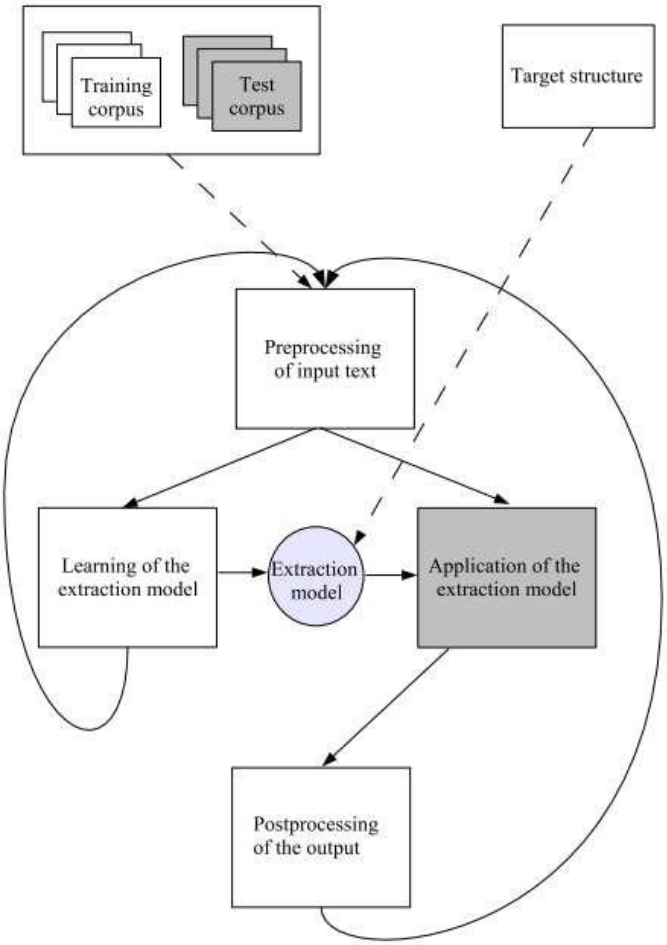
Ekstraksi informasi merupakan suatu proses untuk mengubah informasi tidak terstruktur, semi struktur menjadi data yang terstruktur [1]. Ekstraksi merupakan proses yang bertujuan untuk pemisahan data yang dimana pada kasus ini adalah untuk memisahkan atau menguraikan informasi yang dibutuhkan sesuai dengan kebutuhan.

Sebagai contoh untuk proses ekstraksi informasi yang dilakukan terhadap sebuah *e-mail* mengenai undangan untuk menghadiri perkuliahan. Seperti yang terlihat pada gambar 2.1, hasil dari proses ektraksi informasi tersebut adalah informasi mengenai pembicara, tempat dan watu diadakannya perkuliahan tersebut.

Gambar 2.1 Contoh Hasil Ektraksi Informasi

Terdapat beberapa metode yang dapat digunakan dalam sistem ekstraksi informasi diantaranya metode rule-based, metode statistik dan *knowledge methods* [9]. Pada penelitian ini, ekstraksi informasi akan diterapkan pada dokumen karya tulis ilmiah dengan *HMM* yang dimana algoritma ini menggunakan pendekatan statistik.

Algoritma mempelajari aturan-aturan ekstraksi berdasarkan dokumen teks sebagai data latih yang telah diberi anotasi mengenai entitas informasi yang akan diekstraksi. Peran manusia diperlukan untuk memberikan label atau notasi sesuai entitas yang akan diekstrak pada dokumen. Berikut ini alur proses secara umum sistem ekstraksi informasi dengan algoritma *machine learning* pada Gambar 2.2.



Gambar 2.2 Alur Proses Sistem Ekstraksi Informasi Secara Umum

Alur proses dari gambar 2.2 sebagai berikut [9].

1. *Preprocessing* data masukan

Data masukan berupa teks yang tidak terstruktur, *natural language text*. Informasi penting didapatkan dengan analisis linguistik, karena menghasilkan kata kunci dan fitur ciri penting untuk mengidentifikasi informasi. Analisis linguistik yang digunakan diantaranya tokenisasi, pembagian kalimat (*sentence splitting*), analisis morpologi, *parsing*, dan *named entity*. Pada penelitian ini menggunakan tokenisasi dan ekstraksi fitur termasuk *named entity*.

1.1 Tokenisasi

Keadaan awal dalam bentuk karakter yang terhubung dengan tujuan untuk mengidentifikasi bagian dasar dari *natural language* seperti kata, tanda baca dan

pemisah. Hasil dari token yang memiliki makna dan terhubung sebagai dasar untuk proses linguistik dan teks berikutnya.

1.2 Ekstraksi Fitur

Ekstraksi fitur merupakan proses untuk mencari nilai-nilai fitur yang terkandung dalam dokumen [10]. Fitur dapat diartikan sebagai ciri dari setiap data yang dikenali oleh sistem sehingga menghasilkan nilai fitur. Ekstraksi fitur merupakan topik penting dalam klasifikasi karena fitur-fitur yang baik akan sanggup meningkatkan tingkat akurasi, sementara fitur-fitur yang tidak baik cenderung memperburuk tingkat akurasi [11].

Ada 3 kelompok fitur yang digunakan dengan 15 fitur, yaitu fitur lokal, fitur tata letak dan *named entity*. Fitur lokal adalah karakteristik yang terdapat dalam karekter setiap baris kalimat 7 fitur. Fitur tata letak adalah posisi suatu baris kalimat dalam bagian dokumen 3 fitur. Fitur *named entity* adalah fitur yang diekstrak dari dokumen berdasarkan aturan tertentu 3 fitur [4] sedangkan untuk fitur LOWERCASE dan EIGHTDIGITS merupakan fitur yang ditentukan oleh peneliti.

2. Pembelajaran dan aplikasi model ekstraksi informasi

Algoritma pembelajaran digunakan untuk membentuk model ekstraksi informasi berdasarkan hasil pelatihan data masukan dan penentuan struktur awal seperti anotasi atau label yang telah ditentukan. Hasil pelatihan diaplikasikan ke data pengujian untuk menghasilkan informasi sesuai dengan struktur awal yang ditentukan.

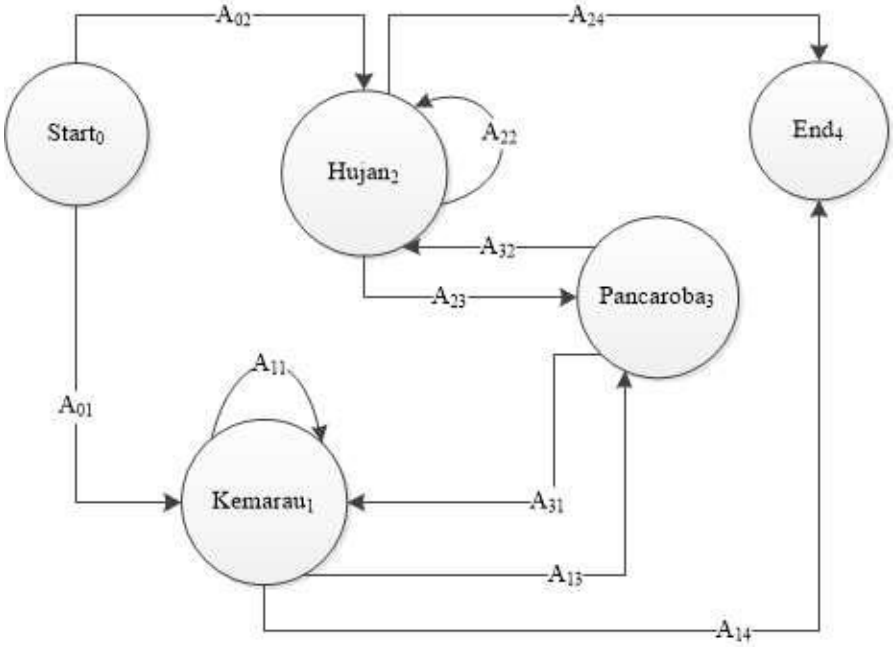
3. *Postprocessing*

Secara umum struktur target direpresentasikan dengan relasi dalam basis data, oleh karena itu pemrosesan hasil keluaran berkaitan dengan mengisi struktur target dengan informasi relevan yang dihasilkan. Proses pengisian tersebut mencakup normalisasi ke dalam format tertentu contohnya untuk representasi tanggal dan waktu. Kemungkinan fakta yang ditemukan dalam dokumen teks dibutuhkan untuk proses penggabungan fakta (*instance unification*).

2.4 Hidden Markov Model

Hidden Markov Model (HMM) merupakan *machine learning* dengan model statistik yang sistemnya diasumsikan sebagai *markovian proccess* yang merupakan bagian dari proses stokhastik yang memiliki properti *Markov*, dan apabila diberikan data masukan keadaan saat ini, keadaan yang akan datang dapat diprediksi dan lepas dari keadaan masa lalu. Dengan kata lain, kondisi masa depan dituju dengan menggunakan probabilitas [12].

HMM merupakan *variant* dari *finite state machine*. *finite state* menerupakan kumpulan state yang transisi antar state-nya dilakukan berdasarkan observasi. Pada *markov chain*, setiap busur antar state berisi probabilitas kemungkinan jalur tersebut akan diambil. Jumlah probabilitas semua busur yang keluar dari sebuah simpul adalah satu.



Gambar 2.3 Contoh Probabilitas Transisi Dari *Hidden Markov Model*

Pada gambar 2.3, A_{ij} menunjukan probabilitas transisi dari state I ke state j. Contoh, simpul $Start_0$ memiliki dua kemungkinan A_{01} dan A_{o2}, sehingga jumlah probabilitas $A_{01} + A_{02} = 1$. Hal ini berlaku juga untuk simpul-simpul yang lain. *Markov Chain* berperan untuk menghitung probabilitas suatu kejadian teramati, dan

untuk mengetahui urutan kejadian yang tersembunyi menggunakan algoritma *HMM*.

Hidden Markov Model diasumsikan dengan kondisi yang tidak terobservasi, dan transisi antara kondisi tidak terlihat secara langsung akan tetapi outputnya dapat dipengaruhi dan bergantung pada keadaan tersebut. Oleh karena itu langkah yang dibuat oleh *HMM* memberikan informasi menunjukan langkah yang dilewati model, bukan kepada parameter dari model tersebut. Walaupun parameter model itu diketahui tetapi model tetap tersembunyi [12].

Parameter pada *HMM* seperti kondisi tersembunyi Q, suatu nilai output observasi O, kemungkinan transisi A, kemungkinan output B, sebuah kondisi awal π. Saat kondisi tidak terobservasi Tetapi, setiap keadaan menghasilkan *output* kemungkinan B. biasanya, Q dan O dimengerti, jadi HMM disebut *triple* (A, B, π) [9].

1. Himpunan observasi state: O = $O_1, O_2, \dots, O_N$.
2. Himpunan hidden state: Q = $Q_1, Q_2, \dots, Q_N$.
3. Probabilitas transisi: A = $a0_1, a0_2, \dots, an_1 \dots an_m$; a_{ij} adalah probabilitas untuk perpindahan dari state I ke state j.
4. Probabilitas emisi atau observasi likehood: B = $b_i(O_t)$, merupakan probabilitas observasi O_t dibangkitkan oleh state ke i.
5. State awal dan akhir: q_0, q_{end}, yang tidak terkait dengan observasi.

2.4.1 Observasi State

Observasi adalah state yang terlihat dan direpresentasikan dengan simbol O [9], pada penelitian ini adalah: Initcaps, Allcaps, Containsdigit, Alldigit, Containsdots, Lowercase, Punctuation, Eightdigit, Word, Line_Start, Line_In, Line_End, Person, Organization, Year.

2.4.2 Hidden State

Hidden adalah kondisi yang tidak terlihat dan direpresentasikan dengan simbol Q [9], pada penelitian ini adalah: Judul Penelitian, Jenis Penelitian, Kalimat Pengajuan, Penulis (sampul), NIM (sampul), Program Studi, Fakultas, Universitas

dan Tahun. Sedangkan untuk kelas pada lembar abstrak seperti Judul Halaman abstrak, Judul Abstrak, Other, Penulis (abstrak), NIM (abstrak), Isi Abstrak, Kata kunci.

2.4.3 Probabilitas Awal (π)

Merupakan probabilitas pemberian nilai sebagai awal dari suatu keadaan yang dimana $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ [9] yang ketika dijumlahkan nilai kondisi awal (π) harus sama dengan 1 [9]. Nilai kondisi awal dibuat sebanyak kelas yang ada.

$$\sum_{j=1}^{n} \pi_1 = 1 \qquad (2.1)$$

2.4.4 Probabilitas Transisi (A)

Merupakan nilai probabilitas dari suatu keadaan untuk berpindah ke keadaan selanjutnya [9]. Probabilitas transisi adalah sebuah matriks yang dilambangkan dengan A = $\{A_{ij}\}$. Jumlah perpindahan tiap kelas didapat dari tabel hasil ekstraksi fitur pada kolom kelas. Berikut persamaannya 2.2 [9]:

$$A_{ij} = \frac{k}{l} \qquad (2.2)$$

Keterangan:

A = Probabilitas Transisi

i = Baris dari matriks

j = Kolom dari matriks

k = Jumlah perpindahan tiap kelas

l = Jumlah total langkah state

2.4.5 Probabilitas Emisi (B)

Nilai emisi didapat dari banyaknya jumlah kemunculan kelas terhadap fitur dibagi dengan seluruh jumlah token. Misalkan perhitungan emisi pada judul penelitian terhadap initcaps, pertama mencari jumlah kemunculan pada tabel ekstraksi fitur dengan kelas = 0 (judul penelitian) dan fitur initcaps = 1 (true/termasuk kedalam fitur initcaps). Berikut persamaannya 2.3 [9]:

$$B_{ij} = \frac{d}{u} \qquad (2.3)$$

Keterangan:

B = Probabilitas Emisi

i = Baris dari matriks

j = Kolom dari matriks

d = Jumlah kemunculan ekstraksi fitur

u = Jumlah Token

2.5 Algoritma Viterbi

Merupakan algoritma *simple dynamic programming* [9], yang digunakan untuk penarikan kesimpulan dan menemukan nilai paling optimal dari permasalahan [13]. Berikut beberapa tahapan dalam V*iterbi*.

2.5.1 Inisialisasi

Pada tahap ini dilakukan untuk mendapat nilai awal pada algoritma viterbi. Berikut persamaannya 2.4 [6].

$$\delta_i = \pi_i * B_i, 1 \leq i \leq N \qquad (2.4)$$

Keterangan:

δ_i = inisialisasi awal

π_i = kondisi awal

B = probabilitas emisi

2.5.2 Rekursi

Pada tahap ini proses perhitungan berulang sebanyak jumlah token kata yang ada. Nilai matriks penyimpanan bernilai 0, karena perhitungan belum dilakukan. Berikut persamaannya 2.5 [7].

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) A_{ij}\right] B_j(O_t), 2 \leq t \leq T, 1 \leq j \leq N \qquad (2.5)$$

$\psi t(j) = 0$

Keterangan:

$\delta_t(j)$ = inisialisasi setiap hasil rekursi

$\max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}]$ = nilai maksimal dari hasil inisialisasi dikali nilai probabilitas transisi

δ_{t-1} = inisialisasi setiap hasil rekursi dikurangi 1

A_{ij} = probabilitas transisi

$b_j\ (O_t)$ = probabilitas emisi

$\psi t(j)$ = matriks penyimpanan hasil rekursi

2.5.3 Terminasi

Terminasi merupakan tahap akhir untuk mendapatkan nilai tertinggi dari proses viterbi. Berikut persamaannya 2.6 [7].

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \qquad (2.6)$$

$$Q_T^* = \arg \max_{1 \le i \le N} [\delta_T(i)]$$

Keterangan:

P^* = probabilitas nilai tertinggi

$\max_{1 \le i \le N} [\delta_T(i)]$ = nilai maksimal dari hasil semua rekursi

Q_T^* = Kondisi tersembunyi

$\arg \max_{1 \le i \le N} [\delta_T(i)]$ = isi dari nilai maksimal

2.5.4 Backtracking

Tahap ini untuk memeriksa ulang nilai terbesar yang telah didapat pada perhitungan rekursi. Berikut persamaannya 2.7 [6]:

$$Q_t^* = \psi_{t+1}(Q_{t+1}^*), t = T-1, T-2, \ldots \qquad (2.7)$$

Keterangan:

Q_t^* = token kata ke-t

$\psi_{t+1}(Q_{t+1}^*)$ = matriks penyimpanan rekursi pada token kata

T = jumlah t dikurangi 1

2.6 Natural Language Processing (NLP)

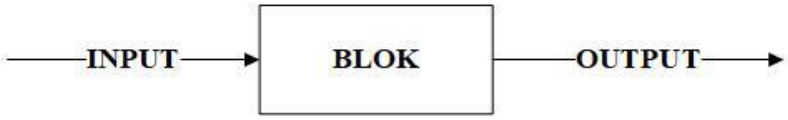
Alat preprocessing yang tepat dalam banyak studi Natural Language Processing (NLP) sangat penting untuk dilakukan memberikan akurasi yang lebih baik. Tahap preprocessing leksikal, seperti pendeteksian kata-kata dasar (Stemming) dan deteksi jenis kata (penandaan POS) berdampak besar bagi sistem komputasi bahasa itu membutuhkan penentuan struktur kalimat. Dalam bahasa Indonesia, penelitian tentang Stemming dan Penandaan POS masih dilakukan, baik dengan menggunakan metode statistik atau aturan tertentu. Beberapa masalah yang dihadapi untuk pengolahan tersebut adalah kurangnya corpus di Indonesia dan Indonesia ketidaklengkapan aturan yang tersedia. Penelitian tentang stemming, pertama kali diterbitkan oleh Julie Beth Lovins pada tahun 1968 [15].

2.7 Pemodelan Sistem

Merupakan pembentukan sebuah model untuk menggambarkan dan memperjelas proses yang terjadi pada suatu sistem. Yang digunakan pada penelitian ini terdiri dari, Blok Diagram, DFD dan Diagram Konteks [16].

2.7.1 Blok Diagram

Merupakan suatu gambaran ringkas dari proses suatu sistem dari gabungan sebab dan akibat antara data masukan, proses dan keluaran yang dihasilkan dari sistem [16].



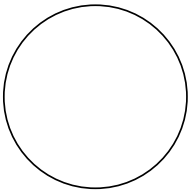
Gambar 2.4 Blok Diagram

Dalam penelitian ini blok diagram digunakan untuk menggambarkan proses yang terjadi pada sistem dari data masukan hingga keluaran.

2.7.2 DFD (Data Flow Diagram)

Merupakan suatu diagram untuk menggambarkan suatu alur proses data pada sistem secara logik, DFD digunakan pada tahap analisis maupun desain. Dalam DFD terdapat 4 simbol utama, dilihat pada Tabel 2.3 [16].

Tabel 2.3 Blok Diagram

Simbol	Keterangan
	**Entitas Eksternal:** menggambarkan asal atau tujuan data.
	**Lingkaran:** menggambarkan proses yang dilakukan sistem maupun manual.
	Aliran Data: menggambarkan aliran masukan maupun keluaran data.
	**Simpanan Data:** digunakan untuk menyimpan data dari sistem ataupun untuk memanggilkan data.

Dalam penelitian ini DFD digunakan untuk menggamabarkan proses yang dilakukan pada sistem ekstraksi informasi dokumen karya tulis ilmiah menggunakan *Hidden Markov Model*.

2.7.3 Diagram Konteks

Merupakan model sistem yang belum dianalisis terhadap proses yang terdapapt pada sistem. Diagram konteks juga termasuk dari bagian Data Flow Diagram (DFD) [16]. Ada beberapa karakteristik pada diagram konteks, yaitu:

1. Hanya memiliki satu proses.
2. Tidak ada penomoran pada setiap prosesnya.
3. Semua arus data digambarkan secara rinci.

Dalam penelitian ini, diagram konteks digunakan untuk menjelaskan tentang alur masukan data, proses dan keluaran data.

2.8 Tokenisasi

Tokenisasi merupakan tahap pemotongan teks masukan yang berupa kumpulan kalimat menjadi kumpulan kata tunggal. Pemotongan teks masukan menjadi kumpulan kata tunggal berdasarkan dengan spasi [13]. Contoh tokenisasi dapat dilihat pada Tabel 2.4.

Tabel 2.4 Contoh Tokenisasi

Teks Masukan	Hasil Tokenisasi
ANALISIS GRAY LEVEL DIFFERENCE METHOD DAN METODE NAIVE BAYES MENGIDENTIFIKASI PENYAKIT LIDAH MANUSIA	ANALISIS
	GRAY
	LEVEL
	DIFFERENCE
	METHOD
	DAN
	METODE
	NAIVE
	BAYES
	MENGIDENTIFIKASI
	PENYAKIT
	LIDAH
	MANUSIA

Tokenisasi kata digunakan pada penelitian ini bertujuan untuk pemisahan kata, angka dan yang mengandung simbol. Untuk memudahakan pada proses ekstraksi fitur dan pembobotan serta untuk pembuatan state pada proses HMM.

2.9 Ekstraksi Fitur

Ekstraksi fitur merupakan proses untuk mencari dan mendapatkan nilai-nilai yang terkandung dalam sebuah dokumen. Fitur diartikan sebagai ciri atau pembeda dari setiap data agar mudah dikenali oleh sistem. Ekstraksi fitur sangat penting dalam proses klasifikasi, karena fitur yang baik akan membuat hasil dari akurasi meningkat, semetara jika fiturnya kurang baik akan memperburuk atau mengurangi tingkat akurasi. Ekstraksi fitur yang digunakan sebanyak 15 fitur

seperti penelitian sebelumnya [4]. Berikut keterangan dari fitur yang digunakan pada penelitian dapat dilihat pada Tabel 2.5.

Tabel 2.5 Ekstrasi Fitur

No	Ekstraksi Fitur	Singkatan Pada Tabel	Keterangan
Fitur Lokal			
1	INITCAPS	IC	Mengenali setiap token yang hurufnya diawali dengan kapital.
2	ALLCAPS	AC	Mengenali setiap token yang semua hurufnya kapital.
3	CONTAINSDIGIT	CDG	Mengenali setiap token yang mengandung angka.
4	ALLDIGIT	AD	Mengenali setiap token yang semuanya angka.
5	CONTAINSDOTS	CDT	Mengenali setiap token yang mengandung titik.
6	LOWERCASE	LC	Mengenali setiap token yang semuanya huruf kecil.
7	PUNCTUATION	PT	Mengenali setiap token yang mengandung tanda tertentu seperti titik, koma, titik dua, semi colon, tanda kurung dan tanda seru.
8	EIGHTDIGIT	ED	Mengenali setiap token yang memiliki digit sebanyak 8 digit.
9	WORD	WR	Fitur khusus untuk memberikan bobot pada token untuk kelas Jenis_Penelitian dan Kalimat_Pengajuan.
Fitur Tata Letak			
10	LINE_START	LS	Mengenali posisi token berada pada awal baris.
11	LINE_IN	LI	Mengenali posisi token berada pada pertengahan baris.
12	LINE_END	LE	Mengenali posisi token berada pada akhir baris.
Fitur Named Entity			
13	PERSON	PR	Mengenali token nama seseorang.
14	ORGANIZATION	ORG	Mengenali token sebuah organisasi.
15	YEAR	YR	Mengenali token tahun.

Setiap token yang termasuk kedalam salah satu fitur di atas akan diberikan bobot 1 dan jika tidak termasuk akan diberikan bobot 0. Pemberian bobot dilakukan karena pada konsep pembelajaran *machine learning* membutuhkan suatu nilai statistik yang dapat dikenali oleh mesin itu sendiri, agar dapat melakukan proses pembelajaran dan perhitungan.

2.10 *PDF*

Portable Document Format (PDF) merupakan format *file* untuk dokumen-dokumen yang dibutuhkan untuk mewakili dokumen yang asli, karena semua elemen yang ada pada dokumen asli disimpan sebagai gambaran elektronik [17].

Dalam penelitian ini format *PDF* didapat setelah melakukan proses scanning dokumen sampul dan abstrak yang berupa *hardcopy*, dan proses ini dilakukan diluar sistem.

2.11 *TIFF (Tagged Image File Format)*

Merupakan format terbaik pada gambar karena semua data dan informasi (data RGB, data CMYK dan lainnya) yang terdapat pada gambar tidak hilang. Format ini biasa digunakan untuk kebutuhan pencetakan dengan kualitas sangat tinggi sehingga ukuran berkas juga sangat besar [18].

Dalam penelitian ini penggunaan format gambar *TIFF* dibutuhkan untuk *library tesseract*. Yang dimana *library* ini menyarankan ketika melakukan proses konversi gambar, format gambar harus memiliki kualitas yang bagus. Dan proses ini dilakukan diluar sistem.

2.12 *TXT*

Plain Text (TXT) merupakan jenis teks murni yang hanya berupa karakter teks tanpa ada format lainnya [19]. *Plain text* pada penelitian ini menghasilkan teks yang tidak memiliki bentuk pada teksnya seperti tipe *font*, warna, ukuran, dan lain-lain. *Plain text* yang tidak memiliki kriteria tipe *font*, warna, ukuran, dihasilkan oleh penggunaan *library Tesseract*.

2.13 *Regular Expression*

Regular Expression (Regex) merupakan suatu bentuk notasi yang digunakan untuk mengolah teks, termasuk mendeskripsikan dan memisahkan teks [20]. Pada penelitian ini *Regex* dibutuhkan pada tahapan ekstraksi fitur. Dengan aturan *Regex* yang dibuat setiap token mendapatkan nilai fitur.

2.14 *CSV*

Comma Separated Values (CSV) merupakan format yang terdiri dari file teks yang berisis *line* dan *value* [10]. Pada *file CSV* terdapat beberapa istilah seperti *Value, Line, Header, Row dan Cell*. *Line* adalah setiap baris *header* yang tidak termasuk sebaggai *row*. *Value* adalah konten pada *cell* untuk *row*. *Cell* merupakan kolom, dan *row* merupakan baris.

Dalam penelitian ini, *file CSV* digunakan untuk keperluan *data training* dan sebagai tempat penyimpanan data, yang bertujuan agar data dapat tersusun dengan rapih terhadap banyaknya data.

2.15 Perancangan Sistem

Perancangan sistem bertujuan untuk menggambarkan, merencanakan, dan pembuatan sketsa dari beberapa elemen terpisah menjadi satu kesatuan sistem yang akan dibangun.

2.16 Bahasa Pemrograman

Bahasa pemrograman merupakan bahasa yang digunakan dalam pembangunan sistem ekstraksi informasi dokumen karya tulis ilmiah menggunakan *Hidden Markov Model*. Bahasa pemrograman yang digunakan adalah PHP (*Personal Home Page*).

2.16.1 *PHP (Personal Home Page)*

PHP merumpakan *server side–scripting* yang menyatu dengan *HTML* untuk mengelola *web* dinamis. Karena *PHP* merupakan *server-side-scripting* maka sintaks dan perintah-perintah *PHP* akan diesksekusi diserver 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* [18]. Dalam penelitian ini menggunakan bahasa pemrograman *PHP*, karena program akan dibuat berbasis *web*.

2.17 Pengujian Akurasi

Pengujian akurasi merupakan tahapan evaluasi proses sistem ekstrasi informasi. Pada penelitian ini, sekumpulan token yang sudah memiliki kelas dari hasil ekstraksi informasi akan dihitungan nilai akurasi [4] terhadap token yang benar pada proses klasifikasi.

2.17.1 Nilai Akurasi

Nilai akurasi didapatkan dengan mengukur nilai token yang benar dari proses klasifikasi, dibagi dengan jumlah token, dikalikan 100%. Berikut rumus perhitungan Nilai akurasi [4].

$$Akurasi\ (\%) = \frac{Seluruh\ data\ terklasifikasi\ dengan\ benar}{Seluruh\ testing\ data}\ x\ 100\% \qquad \textbf{(2.7)}$$

Maksud “Seluruh data terklasifikasi dengan benar” adalah jumlah seluruh kelas yang terklasifikasi dengan benar antara data asli dan data prediksi. Sedangkan untuk “seluruh testing data” merupakan jumlah keseluruhan data yang dijadikan sebagai data untuk diklasifikasikan. Pada penelitian ini, data yang dimaksud ialah token.

2.18 Perangkat Lunak Pendukung

Perangkat lunak pendukung adalah suatu kebutuhan perangkat yang digunakan dalam pembangunan sistem ekstraksi informasi dokumen karya tulis

ilmiah. Perangkat lunak pendukung yang digunakan dalam penelitian ini meliputi *XAMPP* dan *Visual Studio Code*.

2.18.1 *XAMPP*

XAMPP merupakan perangkat lunak yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program MariaDB *database*, *Apache HTTP Server*, dan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi), *Apache*, *MariaDB*, *PHP* dan *Perl* [18].

Pada penelitian ini menggunakan *Xampp Apache* sebagai *web server* yang bekerja untuk melayani *request* dari HTTP *client* (*browser*) untuk menampilkan hasil eksekusi kode php.

2.18.2 *Visual Studio Code*

Visual Studio Code adalah editor kode sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *MacOS*. Ini termasuk dukungan untuk *debugging*, kontrol Git yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi. Ini gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi *proprietary* [18].

Visual Studio Code didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk *desktop* yang berjalan pada tata letak. Meskipun menggunakan kerangka Elektron, perangkat lunak tidak menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode nama "Monaco") yang digunakan dalam *Visual Studio Team Services* (sebelumnya disebut *Visual Studio Online*).

Pada penelitian ini *Visual Studio Code* digunakan untuk penulisan koding pembuatan program sistem ekstraksi informasi dokumen karya tulis ilmiah menggunakan *Hidden Markov Model*.

2.18.3 *Imagemagick*

Imagemagick merupakan *image utilities* yang bekerja dengan *interface command line*. Seluruh kebutuhan seperti konversi suatu format ke format lainnya, menampilkan gambar, menampilkan identifikasi gambar dapat diperoleh dengan perangkat lunak ini [23].

2.18.4 *Tesseract*

Tesseract merupakan free engine Optical Character Recognition (OCR) yang dirilis dibawah lisensi Apache dan pengembangannya disponsori oleh Google. Tesseractsaat ini merupakan salah satu engine OCR open source yang paling akurat dibanding dengan engine yang lain. Tesseract dapat membaca berbagai format gambar dan mengkonversinya ke teks. Selain gambar, Tesseract juga dapat membaca file PDF [24].

OCR tool Tesseract digunakan pada penelitian ini karena mempunyai tingkat akurasi ekstraksi lebih besar dari 42% dan 25 kali lebih cepat untuk waktu pemrosesan ekstraksi dibandingkan dengan tool sebelumnya yaitu OCRopus [24].