BAB 2
TINJAUAN PUSTAKA

2.1. Ekstraksi Informasi

Ekstraksi Informasi adalah pengolahan data pada suatu bidang ilmu yang mana dibutuhkan pengambilan data-data yang berupa teks dan juga data yang beragam. Ekstraksi informasi juga merupakan suatu bidang ilmu untuk pengolahan bahasa alami dengan cara mengubah teks yang tidak terstruktur menjadi informasi teks dalam bentuk yang lebih terstruktur [1].

Ekstraksi informasi diperlukan untuk mendapatkan informasi dari suatu data yang berbentuk teks sehingga dapat digunakan untuk mendapat informasi di dalamnya yang mana sebelumnya telah melalui proses analisis, dan ketegorisasi [1]. Dalam melakukan suatu proses ekstraksi informasi diperlukan data yang cukup banyak sebagai data training agar suatu mesin dapat belajar untuk membuat output yang mampu membuat prediksi yang tepat ketika diuji dengan data testing.

Misalnya, terdapat data teks dari halaman cover seperti berikut “Pembangunan aplikasi auto chapture berdasarkan pergerakan objek pada platform android dengan menggunakan metode sum of absolute different” data tersebut merupakan kategori dari sebuah judul skripsi, mesin pun dapat mengenali berdasarkan ciri dari kategori judul tersebut, sehingga jika ada data yang baru mesin dapat mengenali dan memprediksi yang mana termasuk katagori judul.

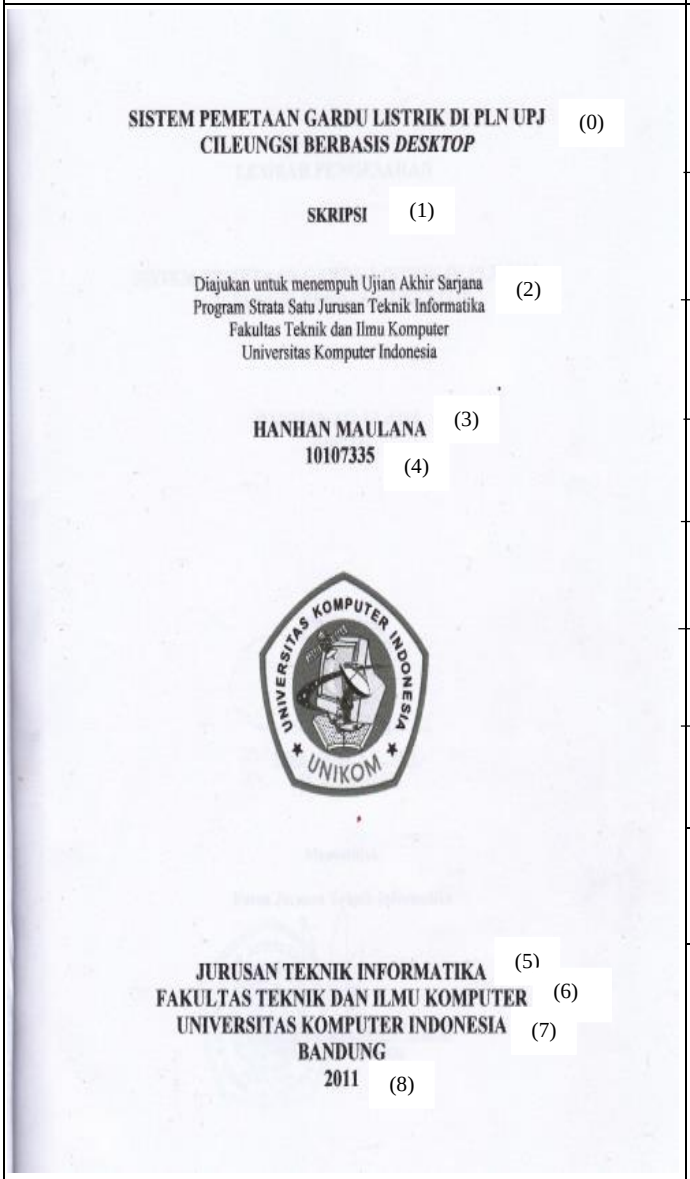
2.2. Karya tulis ilmiah skripsi

Karya tulis ilmiah adalah tulisan atau laporan tertulis yang memaparkan hasil penelitian atau pengkajian suatu masalah oleh seseorang atau sebuah tim dengan memenuhi kaidah dan etika keilmuan yang dikukuhkan dan ditaati oleh masyarakat [2]. Data, simpulan, dan informasi lain yang terkandung dalam karya ilmiah tersebut dijadikan acuan (referensi) bagi ilmuan lain dalam melaksanakan penelitian atau pengkajian selanjutnya [3].

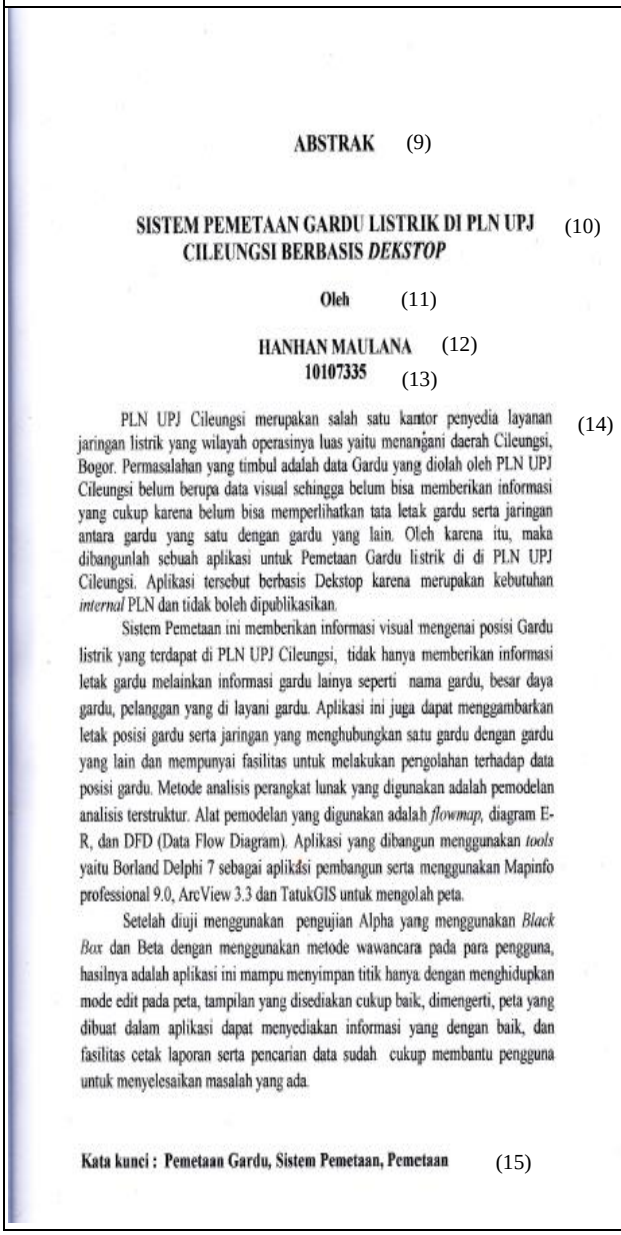
Skripsi merupakan karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademisnya [4] skripsi juga merupakan suatu syarat yang harus dipenuhi oleh mahasiswa/mahasiswi untuk mendapatkan gelar strata 1 atau lebih dikenal dengan S1.

Penelitian yang akan dibuat ini menggunakan karya tulis ilmiah pada lembar cover dan abstrak skripsi sebagai data yang bertujuan untuk mengambil informasi yang ada didalamnya, karena di dalamnya terdapat berbagai kategori yang bermacam-macam. Pada lembar cover memiliki kategori judul penelitian, jenis penelitian, kalimat pengajuan, nama, nim, program studi, fakultas, universitas dan tahun. Sedangkan pada abstrak terdapat judul penelitian, oleh, nama, nim, isi abstrak serta kata kunci.

Tabel 2.1 Bagian-bagian Kategori dari lembar sampul skripsi Tabel

Lembar sampul skripsi	No	Kategori	Kelas
	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 Bagian-bagian kategori dari lembar sampul abstrak

Lembar abstrak Skripsi	No	Ketegori	Kelas
	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

Pada lembar sampul dan abtrak terdapat angka, dimana angka tersebut menunjukan urutan dari bagian yang direpresentasikan menjadi kelas agar mudah diolah datanya untuk proses klasifikasi menggunakan metode GHMM. Disini terdapat

15 kelas yang akan digunakan pada penelitian ini mulai dari kelas judul penelitian, jenis penelitian, kalimat pengajuan, dan lainnya yang terdapat pada tabel di atas.

2.3. Penelitian Terdahulu

Pada penelitian sebelumnya GHMM sudah pernah diterapkan seperti pada web information ekstraksi dengan pengujian sebuah halaman website untuk mengetahui setiap komponen yang ada pada web seperti header, isi, footer dengan akurasi tertinggi yang di dapatkan yaitu 85,5% dengan menggunakan urutan dari perpindahan transisi [1].

Pada penelitian sebelumnya pada kasus pengenalan Gen manusia dalam DNA di identifikasi akurasi tertinggi hingga 85% dari basis pengkodean protein yang benar yaitu sebesar 80%, serta 58% ekson persis di identifikasi dengan spesifisitas sebesar 51% [8].

Pada penelitian sebelumnya yaitu GHMM untuk koreksi ejaan, yang mana terdapat 2 kategori ejaan yang salah, yaitu in-word kata terdapat pada kamus tetapi pada pengejaannya ada kata yang hilang ataupun kata yang berlebih, sedangakan kesalahan cross-word adalah kata yang seharusnya menjadi satu kesatuan namun terpisah sehingga menjadi 2 buah kata yang berbeda, dan didapatkan GHMM memiliki tingkat yang lebih baik dibanding metode Noisy channel model [5].

2.4. Algoritma

Pada pembuatan suatu sistem untuk membuat alur ataupun rangkaian dari suatu proses sistem, diperlukan pembuatan suatu algoritma. Algoritma menurut Donal E. Knuth, adalah sekumpulan aturan-aturan berhingga yang memberikan sederetan operasi-operasi untuk menyelesaikan suatu jenis masalah yang khusus [1]. Sementara menurut rinaldu munir, algoritma adaah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis [4]. Notasi algoritma biasanya dapat diterjemahkan kedalam berbagai macam bahasa pemrograman.

Berikut adalah penulisan notasi algoritma:

1. Deskriptif

Adalah suatu algoritma yang bentuknya berupa kalimat untuk menggambarkan langkah-langkah dalam penyelesaian suatu masalah tetapi memiliki kekurangan yaitu sulit untuk diterjemahkan kedalam bahasa pemrograman.

2. Pseudocode

Merupakan kode yang mirip dengan bahasa pemrograman yang sebenarnya, didalam penulisannya tidak ada aturan baku, dan biasanya ditulis berdasarkan bahasa pemrograman masing-masing sesuai bahasa pemrograman yang digunakan.

3. Flowchart

Merupakan Algoritma yang kegiatannya digambarkan dalam bentuk simbol, flowchart juga dikenal sebagai (bagan alir). Dengan flowchart dapat dimudahkan untuk melihat langkah-langkah pada proses secara detail.

2.5. Pemodelan Sistem

Dibutuhkan proses-proses yang berguna untuk memberikan gambaran serta pemahaman yang terjadi pada setiap prosesnya.

Berikut adalah pemodelan sistem yang terdiri dari bagian-bagian yaitu Blok Diagram, DFD, Diagram konteks, dan Flowchart

2.5.1. Blok Diagram

Blok Diagram adalah suatu proses yang menggambarkan serta ringkasan dari suatu gabungan yang terjadi oleh sebab dan akibat yang diterima dari data masukan dan keluaran. Blok Diagram juga merupakan rangkain yang dapat menggambarkan proses yang terjadi didalam sistem mulai dari awal input dan outputnya.

Gambar 2.1 Blok Diagram Secara Umum

2.5.2. DFD

Pada pembuatan suatu sistem dibutuhkan pembuatan suatu model yang dapat menggambarkan sistem sebagai suatu jaringan yang mana tiap proses fungsionalnya dihubungkan satu sama lain dengan alur data, baik secara manual ataupun secara komputerisasi.

Umumnya komponen yang terdapat pada DFD yaitu Terminator, Proses, Data Store, dan Alur Data. Berikut adalah fungsi dari komponen yang tersebut :

1. Terminator

   Merupakan bagian dari luar sistem yang menunjukan hubungan sistem dengan dunia luar sistem.

2. Proses

   Menunjukan kegiatan yang akan dilaksanan oleh sistem. Terjadi proses/kegiatan yang berkaitan dengan pengolahan dari masukan yang masuk kedalam sistem yang mana terdapat input dan output yang dihasilkan.

3. Data Store

   Berkaitan dengan penyimpanan data yang dilakukan secara komputerisasi. Data yang ditampung merupakan hasil dari proses.

4. Alur Data

   Menunjukan arah atau alur data yang akan dilanjutkan ketahapan selanjutnya. Alur data berguna untuk menerangkan perpindahan data ataupun informasi dari suatu bagian dari sistem ke bagian lainnya.

2.5.3. Diagram Konteks

Berfungsi untuk memberi gambaran serta analisis secara menyeluruh dan dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Diagram konteks sama sekali tidak memuat penyimpanan data, hanya terdapat proses didalamnya. Berikut adalah karakteristik yang dimiliki oleh Diagram Konteks yaitu:

1. Tidak adanya suatu penomoran khusus yang diberikan pada setiap prosesnya.
2. Diagram konteks tidak memuat penyimpanan data tetapi hanya proses.
3. Seluruh arus data digambarkan secara jelas.

2.5.4. Flowchart

Flowchart merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem.

Flowchart dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Flowchart digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. Dengan begitu setiap urutan proses dapat digambarkan menjadi lebih jelas.

2.6. Tokenisasi

Suatu pemotongan atau pemisahan string kata dari setiap kalimat yang akan dipecah menjadi kata-kata tunggal. Tokenisasi juga digunakan untuk memisahkan antara kata, angka, maupun spasi yang ada pada tiap kalimatnya.

Pada penelitian ini tokenisasi akan dilakukan pada tahapan filtering untuk membuang bagian-bagian yang tidak penting dalam sebuah kalimat tanpa mempertimbangkan duplikasi dari kata tersebut. Contoh tokenisasi dapat dilihat pada tabel 2.3.

Tabel 2.3 Contoh Tokenisasi

Teks masukan	**Hasil Tokenisasi**					
SISTEM PEMETAAN GARDU LISTRIK DI PLN UPJ CILEUNGSI BERBASIS DEKSTOP	SISTEM	PEMETAAN	GARDU	LISTRIK	DI	PLN
	UPJ	CILEUNGSI	BERBASIS	DEKSTOP		

2.7. Ekstraksi Fitur

Adalah suatu pemberian bobot nilai pada fitur yang terkandung pada setiap token kata dari hasil tokenisasi .

ekstraksi fitur pada penelitian ini adalah untuk memberikan ciri khusus pada suatu token kata sehingga nantinya dapat dikenali oleh sistem dan akan diberikan nilai bobot pada kata tersebut. Pemilihan fitur yang tepat dapat meningkatkan akurasi dalam pelabelan.

Ekstraksi fitur yang digunakan pada penelitian ini sebanyak 15 buah fitur, merujuk pada penelitian ekstraksi informasi dokumen karya tulis ilmiah menggunakan LVQ oleh Firdamdam [3].

Berikut adalah fitur-fitur yang ada pada penelitian yang sedang dikerjakan.

Tabel 2.4 15 Macam Ekstraksi Fitur

No	Ekstraksi Fitur	Keterangan
1	INITCAPS	Mengenali setiap token yang hurufnya diawali dengan kapital.
2	ALLCAPS	Mengenali setiap token yang semua hurufnya kapital.
3	CONTAINSDIGIT	Mengenali setiap token yang mengandung digit.
4	ALLDIGIT	Mengenali setiap token yang semuanya digit.
5	CONTAINSDOTS	Mengenali setiap token yang mengandung titik.
6	LOWERCASE	Mengenali setiap token yang semuanya huruf kecil.
7	PUNCTUATION	Mengenali setiap token yang mengandung tanda tertentu seperti titik, koma, titik dua, titik koma, tanda kurung, dan tanda seru.

8	EIGHTDIGIT	Fitur tambahan pada penelitian ini, fitur ini dikhususkan untuk mengenali token yang memiliki digit dengan panjang 8 digit.
9	WORD	Fitur tambahan pada penelitian ini, fitur ini dikhususkan untuk memberikan bobot pada token untuk kela JENIS_PENELITIAN dan KALIMAT_PENGAJUAN.
10	LINE_START	Mengenali posisi token pada indeks array awal.
11	LINE_IN	Mengenali posisi token pada indeks array tengah.
12	LINE_END	Mengenali posisi token pada indeks array akhir.
13	PERSON	Mengenali token nama seseorang.
14	ORGANIZATION	Mengenali token sebuah organisasi.
15	YEAR	Mengenali ciri token tahun.

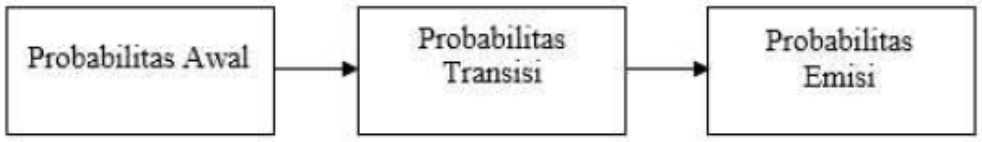
Setiap fitur yang ada akan diberi nilai bobot 1 atau 0, jika kata tersebut termasuk salah satu fitur diatas akan diberi bobot 1, jika tidak akan diberi bobot 0.

2.8. Generalized Hidden Markov Model

Generalized HMM termasuk pengembangan dari HMM (hidden markov model). Generalized hidden markov model sebuah metode yang berbasis probabilitas interval yang mana probabilitas interval tersebut digunakan untuk menangkap ketidakpastian dari perhitungan yang dihasilkan oleh HMM biasa. Hasil yang diberikan

oleh GHMM pada beberapa kasus yang kompleks dapat meningkatkan tingkat akurasi yang dihasilkan karena pada GHMM dilakukan penggabungan kesalahan yang ada selama pengumpulan data untuk tahap pembelajarannya dan pengenalan fitur.

Tahapan training pada GHMM membutuhkan data masukan kelas dan fitur dan token kata. Awal permulaan dari training yaitu mencari parameter dari GHMM, terdapat 3 parameter utama yang harus diketahui nilainya terlebih dahulu yaitu yang termasuk kedalam parameter dari **λ = (A, B, π).** Secara umum langkah-langkah pada prosesnya digambarkan seperti dibawah ini.



Gambar 2.2 Skema Probabilitas GHMM

Berikut adalah tahap-tahap yang ada pada proses training GHMM.

1. Probabilitas Awal

   Probabilitas awal adalah suatu peluang yang terdapat pada suatu keadaan tertentu sebagai awal dari suatu keadaan. Dimana π_i merupakan hasil nilai dari probabilitas q_i yaitu kondisi tersembunyi pada kata ke i, dan S_i jumlah keseluruhan kelas yang ada sampai kelas ke i. Berikut persamaan pada 2.1 [6] :

$$\pi_i = \mathrm{P}\,(q_i = S_i) \qquad (2.1)$$

   Dimana :

   π_i = Probabilitas awal dari kata di awal kelas ke $-i$

   i = Kata di awal kelas ke-i

   P = Probabilitas

   q_i = Kondisi tersembunyi pada kata di awal kelas ke-i

   S_i = Jumlah keseluruhan kelas sampai ke-i

2. Probabilitas Transisi

Adalah Probabilitas dari peluang suatu keadaan bila berpindah dari kelas pertama ke kelas berikutnya. q_j adalah kondisi kelas selanjutnya yang tersembunyi pada saat q_i kondisi kelas saat ini. Probabilitas Transisi dapat dilambangkan dengan A={aij}. Berikut persamaan pada 2.2 [6] :

$$a_{ij} = P\,(q_j \text{ Pada } t+1 \mid q_i \text{ Pada } t) \quad (2.2)$$

Dimana :

a_{ij}	=	Probabilitas transisi dari *i* ke *j*
i	=	Adalah kelas saat ini
j	=	Adalah keadaan selanjutnya
t	=	Merupakan state yang akan di lalui
P	=	Adalah Probabilitas
q_j	=	Kondisi tersembunyi pada keadaan selanjutnya
q_i	=	Kondisi tersembunyi pada keadaan saat ini
t+1	=	Adalah urutan indeks ke-t+1

3. Probabilitas Emisi

Adalah Probabilitas dari suatu peluang distribusi dari keadaan observasi yang mana menghasilkan suatu keadaan baru. Emisi didapatkan dari jumlah kemunculan fitur yang ada pada kelas dan dibagi dengan jumlah keseluruhan dari token kata yang mengandung fitur yang dicari. Misalkan pada emisi akan dilihat kondisi dari tiap token kata yang ada dari kelas ke 1 sampai 16, apakah termasuk kedalam allcaps jika termasuk maka diberi nilai bobot 1, lakukan pengecekan sampai kelas awal telah semua diproses dan dijumlahkan lalu dibagi dengan jumlah token yang mengandung allcaps pada semua kelas.

Di dalam GHMM probabilitas emisi dilambangkan dengan B= ((b(j)) (O(t))). Berikut persamaan 2.3 [6]:

$$b_j\,(O_t\,) = P(\,O \text{ pada } t \mid q_j \text{ pada } t) \qquad (2.3)$$

Dimana :

$b_j\,(O_t\,)$	=	Probabilitas emisi ke –j pada keadaan observersi Ke-t
P	=	Adalah probabilitas
O	=	Adalah kondisi observasi
t	=	Adalah indeks kata ke -t
q_j	=	Adalah fiturnya

4. Observasi Sequence, upper dan lower

Pada Observasi Sequence akan dihitung nilai keseluruhan dari hasil probabilitas awal, transisi dan juga emisi. λ merupakan penampung dari nilai yang telah dicari sebelumnya yaitu pada π, B, A pada persamaan 2.1, 2.2, dan 2.3. Sedangkan O adalah observasi sequence peluang munculnya barisan observasi untuk π, B, A tersebut, dapat dilihat pada persamaan 2.4 [6]. Observasi upper merupakan batas atas dari element yang terkandung dari hasil pengamatan λ yang merupakan nilai dari probabilitas awal, transisi dan juga emisi, sedangkan observasi lower merupakan batas bawah dari element yang terkandung dari hasil pengamatan λ, dapat dilihat pada persamaan 2.5, 2.6 [6] :

$$\text{Observation Sequence } P\,(O \mid \lambda\,) \qquad (2.4)$$

$$\text{Observation Sequence upper } (\bar{O} \mid \lambda\,) \qquad (2.5)$$

$$\text{Observation Sequence lower } (\underline{O} \mid \lambda\,) \qquad (2.6)$$

5. Generalisasi Probabilitas Transisi

Generalisasi probabilitas transisi disini untuk mencari nilai batas bawah dan batas atas dari element yang terkandung pada hasil yang didapatkan dari perhitungan probabilitas transisi dari persamaan 2.2 [6]. Adapun persamaan untuk lower dan upper pada 2.7 dan 2.8 [6] :

$$\underline{A} = \underline{a_{ij}} = p\left(q_{t+1} = S_j \mid q_t = S_i\right)\ (1 \le i,j \le N) \quad (2.7)$$
$$a_{ij} \ge 0, \sum_{j=1} a_{ij} = 1\ (1 \le i,j \le N)$$

$$\overline{A} = \overline{a_{ij}} = p\left(q_{t+1} = S_j \mid q_t = S_i\right)\ (1 \le i,j \le N) \quad (2.8)$$
$$a_{ij} \ge 0, \sum_{j=1} a_{ij} = 1\ (1 \le i,j \le N)$$

Dimana :

$\overline{a_{ij}}$ = Nilai generalisasi upper dari probabilitas transisi

$\underline{a_{ij}}$ = Nilai generalisasi lower dari probabilitas transisi

q_t = Waktu hidden state +1

S_j = Adalah kelas selanjutnya

S_i = Adalah kelasnya

6. Generalisasi Probabilitas Emisi

Generalisasi probabilitas emisi untuk mencari nilai batas bawah dan batas atas dari element yang terkandung pada hasil didapatkan dari perhitungan probabilitas emisi dari persamaan 2.3 [6]. Adapun persamaan untuk lower dan upper pada 2.9 dan 2.10 [6] :

$$\underline{B} = \underline{Bj}\ (k) = p\ (o_t = v_k \mid q_t = S_j) \quad (2.9)$$

$$\overline{B} = \overline{Bj}\ (k) = p\ (o_t = v_k \mid q_t = S_j) \quad (2.10)$$

Dimana :

$\underline{Bj}$ (K) = Probabilitas lower emisi ke –j pada keadaan observersi Ke-t

$\overline{Bj}(k)$ = Probabilitas upper emisi ke –j pada keadaan observersi Ke-t

P = Adalah probabilitas upper

0 = Adalah kondisi observasi

V = Adalah fitur yang terdapat pada kelasnya

Vk = Adalah fitur selanjutnya

7. Generalisasi Probabilitas Awal

Generalisasi probabilitas awal disini untuk mencari nilai batas bawah dan batas atas dari element yang terkandung pada hasil yang didapatkan dari perhitungan probabilitas awal dari persamaan 2.1 [6]. Adapun persamaan untuk lower dan upper pada 2.11 dan 2.12 [6] :

$$\overline{\pi_i} = \overline{P}\,(\,q_1 = S_i\,)\,(\,1 \leq i \leq N) \qquad (2.11)$$

$$\underline{\pi_i} = \underline{P}\,(\,q_1 = S_i\,)\,(\,1 \leq i \leq N) \qquad (2.12)$$

Dimana :

$\overline{\pi_i}$ = Probabilitas awal upper

$\underline{\pi_i}$ = Probabilitas awal lower

P = Probabilitas

q_i = Kondisi tersembunyi pada kata di awal kelas ke-i

S_i = Jumlah keseluruhan kelas sampai ke-i

8. Viterbi

Algorimta viterbi umumnya digunakan untuk menyelesaikan masalah yang berkaitan dengan pengkodean, dapat juga untuk menyelesaikan bidang yang lain. Algoritma viterbi pada GHMM berguna untuk menemukan nilai barisan hidden state dari suatu barisan observasi yang nilainya paling maksimal atau optimal. Pada viterbi terdapat bebeapa tahapan, berikut adalah tahapan pada viterbi.

1. Inisialisasi

Tahapan pertama yaitu inisialisasi. Pada tahap ini akan dicari nilai awal yang dihasilkan dari viterbi. Persamaan yaitu pada 2.13 dan 2.14 [6]

Inisialisasi lower dirumuskan dengan :

$$\delta_1^l(i) = \pi_i b_i(\underline{o_1}) \ (1 \leq \text{i} \leq \text{N}) \tag{2.13}$$

Inisialisasi upper dirumuskan dengan :

$$\delta_1^u(i) = \pi_i b_i(\bar{o_1}) \ (1 \leq \text{i} \leq \text{N}) \tag{2.14}$$

$$\psi_1(i) = 0 \ (1 \leq \text{i} \leq \text{N})$$

Dimana :

$\delta_1^l(i)$	=	Adalah kondisi inisialisasi awal lower
$\delta_1^u(i)$	=	Adalah kondisi inisialisasi awal upper
π_i	=	Adalah probabilitas awal
b_i	=	Adalah probabilitas emisi
$\underline{O}$	=	Observasi lower
$\bar{O}$	=	Observasi upper

2. Induksi

Pada tahapan induksi akan dihitung hasil yang didapat dari inisialisasi, perhitungan dilakukan sebanyak kelas yang ada. Berikut adalah rumus mencari induksi. Berikut persamaannya 2.15 dan 2.16 [6] :

Induksi lower dirumuskan dengan :

$$\delta_1^1(j) = \max_{1 \le i \le N} \{\delta_{t-1}^1(i) a_{ij}\}\, b_j(\underline{o_t}) \quad (2 \le t \le T,\ 1 \le j \le N) \qquad (2.15)$$

Induksi upper dirumuskan dengan :

$$\delta_1^u(j) = \max_{1 \le i \le N} \{\delta_{t-1}^u(i) a_{ij}\}\, b_j(\overline{o_t}) \quad (2 \le t \le T,\ 1 \le j \le N) \qquad (2.16)$$

$$\psi_1(j) = argmax \left\{ \min_{1 \le i \le N} (pro\ \delta_{t-1}^1(i) a_{ij}),\ \min_{1 \le i \le N} (pro\ \delta_{t-1}^u(i) a_{ij} \right\}$$

Dimana :

$\delta_1^l(j)$	=	Inisialisasi setiap hasil induksi lower
$\delta_1^u(i)$	=	Inisialisasi setiap hasil induksi upper
a_{ij}	=	Adalah probabilitas transisi
b_i	=	Adalah probabilitas emisi
$pro\ \delta_{t-1}^1$	=	Penginisialisasian hasil dari induksi dikurangi 1
$\psi_1(j)$	=	Menyimpan hasil induksi

3. Terminasi

Pada tahapan terminasi akan dilakukan untuk mendapatkan hasil atau nilai yang terbaik. Berikut adalah rumus mencari terminasi. Berikut persamaan pada 2.17 [6].

$$P^* = \max_{1 \le i \le N} \{\delta_T^l(i), \delta_T^u(i)\}, \quad (2.17)$$

$$q_T^* = \operatorname{argmax} \left\{ \min_{1 \le i \le N} (pro\delta_T^l(i)), \min_{1 \le i \le N} (pro\delta_T^u(i)) \right\} ,$$

Dimana :

P^* = Probabilitas dari nilai yang terbaik

$\max_{1 \le i \le N}$ = Hasil nilai maksimal dari induksi

q_T^* = Kondisi yang tersembunyi

argmax = Isi yang dihasilkan dari nilai maksimal

4. Backtracking

Pada tahapan backtracking, akan di cek kembali urutan yang terbaik yang dihasilkan dari terminasi. Berikut adalah perhitungan untuk Backtracking pada persamaan 2.18 [6].

$$Q_t^* = \psi_{t+1}(Q_{t+1}^*), t = T-1, T-2, \ldots, 1 \quad (2.18)$$

Dimana :

Q_t^* = Kondisi Tersembunyi

$\psi_{t+1}(Q_{t+1}^*)$ = Matriks Penyimpanan state tersembunyi

q_T^* = Kondisi yang tersembunyi

T = Jumlah T dikurangi 1

2.9. TXT

Format ".txt" umum dapat dibuka di semua komputer, hampir tanpa memerlukan software khusus karena biasanya sudah disediakan dari komputer. Pada penelitian ini file dengan format txt akan digunakan pada tahapan preprocessing pada data testing.

2.10. CSV

CSV bisa dipisahkan dengan menggunakan koma (,) atau titik koma (;). Yang kita perlu lakukan hanyalah menyisipkan tanda titik koma di antara data-data yang ada. Dan kita bisa lakukan dengan find and replace. Biasanya dengan shortcut ctrl+H. tapi terlebih dahulu kita bersihkan data-data yang tidak kita perlukan.

2.11. Regular Expression

Regular Expressions (Regex) merupakan suatu metode yang sangat baik untuk memanipulasi data text. Pada penelitian ini Regex digunakan pada proses ekstraksi fitur untuk memberikan bobot pada data berbentuk text dengan cara membuat aturan Regex. Adapun simbol-simbol yang dapat digunakan untuk pembuatan aturan Regex dipaparkan pada tabel dibawah sebagai berikut.

Tabel 2.5 Macam Regular Expression

Simbol	Fungsi
/	mengawali dan mengakhiri (mengapit) *pattern*
^	mencocokkan *pattern* yang terletak pada awal subjek
$	mencocokkan *pattern* yang terletak pada akhir subjek
.	mencocokkan dengan karakter apapun, kecuali baris baru
.*?	mencocokkan dengan karakter apapun termasuk baris baru
[ ]	membuka dan menutup definisi *character class*
|	tanda pemisah dari untuk opsi alternatif
( )	membuka dan menutup sub-*pattern*
\	karakter *escape*
{x, y}	pembilang repetisi dengan nilai minimal x dan maksimal y
?	pembilang repetisi minimal nol dan maksimal satu {0, }
*	pembilang repetisi minimal nol dan maksimal tidak terbatas {0, 1}
+	pembilang repetisi minimal satu dan maksimal tidak terbatas {1, }

2.12. Perangkat Lunak Pendukung

2.12.1. MySQL

MySQL adalah database server open source yang cukup popular keberadaanya. Dengan berbagai keunggulan yang dimiliki, membuat software database ini banyak digunakan oleh praktisi untuk membangun suatu project. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh Mysql, memungkinkan bermacam-macam aplikasi Komputer yang ditulis dengan berbagai bahasa pemograman dapat mengakses basis data MySQL.

MySQL juga merupakan brand yang terpopuler yang banyak digunakan dari software RDBMS. MySQL membuat database untuk menyimpan dan memanipulasi data, serta menentukan keterkaitan antara masing-masing tabel.

2.12.2. Xampp

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat mendownload langsung dari webresminya.

2.12.3. Atom

Atom adalah sebuah text editor yang memiliki lisensi open source yang tersedia untuk platform OS X, Linux dan Windows. Atom ini dibuat oleh GitHub dan di klaim sebagai text editor yang bisa di custom dengan merubah file configurasinya. Atom adalah text editor yang mirip dengan sublime text. Namun fiturnya lebih lengkap dan mudah untuk digunakan. Atom sebagai editor menyediakan berbagai macam fitur mulai dari package, themes, costumize, dan open source.

2.13. Nilai Akurasi

Nilai akurasi diperoleh dengan mengukur nilai kebenaran token yang diklasifikasikan, dibagi dengan jumlah token, dikali 100%. Berikut rumus perhitungan Nilai akurasi [3].

$$Akurasi\ (\%) = \frac{Keseluruhan\ data\ terklasifikasi\ dengan\ benar}{Keseluruhan\ testing\ data}\ x\ 100\%$$

Rumus diatas menjelaskan bahwa "Keseluruhan data terklasifikasi dengan benar" merupakan jumlah keseluruhan kelas yang terklasifikasikan dengan benar antara data sesungguhnya dan data prediksi. Sedangkan untuk "Keseluruhan testing data", merupakan jumlah keseluruhan data yang dijadikan sebagai data untuk diklasifikasikan. Pada penelitian ini, yang dimaksud dengan data merupakan token. Jadi, pengukuran akan dilakukan terhadap token yang telah memiliki kelas. Dari hasil keseluruhan yang terklasifikasi dengan benar maka jumlahnya akan dibagi dengan seluruh data testing dan dikali dengan 100%, nantinya akan didapatkan jumlah akurasi benar dalam bentuk %.

2.14. Nilai Error

Nilai error merupakan nilai yang diperoleh ketika beberapa kelas tidak terklasifikasi dengan benar. Nilai dari data tidak terklasifikasi dengan benar dibagi dengan jumlah seluruh token dan dikali dengan 100%. Berikut rumus perhitungan error [3].

$$Error\ (\%) \quad = \frac{Keseluruhan\ data\ tidak\ terklasifikasi\ dengan\ benar}{Keseluruhan\ testing\ data}\ x\ 100\%$$

Rumus diatas menjelaskan bahwa “Keseluruhan data tidak terklasifikasi dengan benar” merupakan jumlah keseluruhan data yang tidak terklasifikasikan dengan benar dari perbandingan antara data sesungguhnya dan data prediksi. Sedangkan untuk “Keseluruhan testing data”, merupakan jumlah dari keseluruhan data yang dijadikan sebagai data untuk diklasifikasikan.

2.15. Natural Language Processing (NLP)

Natural language processing atau yang biasa dikenal NLP banyak sekali tools nya. NLP pada banyak kasus sangat penting untuk mendapatkan akurasi yang baik. Seperti pada pendeteksian kata-kata dasar stemming dan deteksi jenis kata (Penandaan PO). Pada banyak kasus penggunaan NLP banyak di lakukan salah satunya penelitian tentang stemming dan penandaan POS. dalam penenganannya NLP terkadang mengalami masalah salah satunya pada kasus di atas adalah kurangnya corpus di Indonesia serta tidak lengkapnya aturan yang tersedia penelitian tentang stemming, pertama kali diterbitkan oleh Julia Beth Lovins pada tahun 1968 [13].