BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam komunikasi sehari-hari, ada dua macam bahasa yang banyak digunakan masyarakat di Indonesia, yaitu bahasa formal dan bahasa tidak formal. Bahasa formal merupakan bahasa yang mengikuti kaidah bahasa, sedangkan bahasa yang tidak formal merupakan bahasa yang keluar dari kaidah buku bahasa [1].

Bukit Senyum merupakan suatu daerah yang terletak di Batu Ampar Provinsi Kepulauan Riau, Kota Batam, Indonesia. Di Daerah Bukit Senyum masih banyak masyarakat sekitar yang menggunakan Bahasa Jawa ngoko dalam kehidupan sehari-hari, baik dalam jual beli dan acara adat istiadat. Walaupun masyarakat sekitar sudah terbiasa mendengar dan mengucapkan Bahasa Jawa Ngoko masih banyak masyarakat yang belum mengetahui penulisan maupun ejaan Bahasa Jawa Ngoko, sehingga masyarakat Bukit Senyum sering salah dalam penulisan Bahasa Jawa ngoko kegunaannya dalam kehidupan sehari hari.

Selain itu kurangnya infomasi tentang kosa kata bahasa Jawa Ngoko yang tepat sehingga banyak kalangan anak muda maupun orang tua sekarang yang tidak tau tulisan baku bahasa Jawa Ngoko, mengatasi hal tersebut dibutuhkan suatu teknologi baru yang dapat mendeteksi kesalahan ejaan penulisan salah satunya teknologi dalam bidang *Natura Language Processing* (NLP) merupakan salah satu cabang ilmu *Artificial Intelligence* (AI) yang berfokus pada pengolahan bahasa natural atau bahasa yang secara umum digunakan oleh manusia untuk

berkomunikasi satu sama lain, dan ada juga teknologi *Normalisasi Teks* (NT) yang berfungsi untuk mengubah teks kalimat menjadi teks yang secara lengkap memperlihatkan cara pengucapannya.

Levenshtein Distance adalah sebuah matriks untuk mengukur angka perbedaan antara 2 *string*, jarak antara *string* diukur berdasarkan angka penambahan karakter, penghapusan karakter ataupun penggantian karakter yang diperlukan untuk mengubah *string* sumber menjadi *string* [2].

Penelitian dengan metode *Levenshtein Distance* ini pernah diteliti Rachmania Nur Dwitiyastuti dengan judul " Pengoreksi Kesalahan Ejaan Bahasa Indonesia dengan Menggunakan Metode *Levenshtein Distance*", dari hasil penelitian yang telah dilakukan dapat ditarik kesimpulan berdasarkan hasil pengujian pencarian kata sistem bisa mendeteksi 100 % kesalahan kata, dan juga dari hasil pengujian sudah bisa mendeteksi kesalahan dan memberi sugesti untuk tiap kategori kesalahan ejaan. Untuk membuat pengoreksian dalam sistem pengoreksi kesalahan ejaan ini lebih tepat, kosa kata dalam kamus sebaiknya dilengkapi lagi [3].

Penelitian lainnya dengan judul "Deteksi Konten *Hoax* Berbahasa Indonesia Pada Media Sosial Menggunakan Metode *Levenshtein Distance*" yang diteliti oleh Frista Gifti Weddiningrum, dari hasil penelitian yang telah diteliti dapat diambil kesimpulan bahwa penerapan metode *Levenshtein Distance* yang dipadukan dengan TF-IDF terbukti mampu membedakan antara berita *hoax* dan tidak, dengan tingkat akurasi yang bagus, dengan 40 berita sebagai data uji dengan pembagian 20 berita *non-hoax* dan 20 berita *hoax* diperoleh hasil akurasi

70% yang berarti semakin banyak kata *hoax* yang dijadikan data latih, maka semakin akurat sistem melakukan pendeteksian [4].

Sesuai yang tertulis di latar belakang di atas, maka penulis mengangkat judul Normalisasi Kata Bahasa Jawa Ngoko menggunakan algoritma *Levenshtein Distance* dengan tingkat akurasi *Suggestion Adequacy*, dengan tujuan mengubah kata yang mengalami kesalahan penulisan menjadi kata baku dengan menampilkan rekomendasi kata yang memiliki jumlah jarak kemiripan terkecil. Aplikasi yang dibangun diharapkan mempunyai tingkat akurasi yang tinggi dan dapat mempengaruhi tahapan penelitian selanjutnya seperti pada proses klasifikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, rumusan masalah yang dapat diambil yaitu "Bagaimana menerapkan algoritma *Levenshtein Distance* dalam normalisasi kata Bahasa Jawa ngoko dan juga penggunaan tingkat akurasi *Suggestion Adequacy*?".

1.3 Batasan Masalah

Supaya pembahasan dalam penelitian ini tidak menyimpang dari pembahasan maka perlu dilakukan batasan masalah. Adapun batasan masalahnya yaitu :

1. Data yang akan digunakan yaitu kata dari Bahasa Jawa ngoko di Bukit Senyum.
2. Bahasa Jawa ngoko yang digunakan dengan mengetik perkalimat dan pengecekannya perkata.

3. Data acuan (kamus) bersumber dari sebuah buku kamus bahasa jawa oleh Joko Sukoyo., M.Pd.
4. Aplikasi yang dibangun berbasis *web* dengan menggunakan bahasa pemograman PHP dan MySQL sebagai *database*.
5. Metode yang digunakan dalam normalisasi Bahasa Jawa ngoko adalah metode *Levenshtein Distance*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Menerapkan algoritma *Levenshtein Distance* dalam normalisasi kata Bahasa Jawa ngoko.
2. Membuat aplikasi normalisasi kata berbasis *web* dengan menggunakan metode *Levenshtein Distance*.
3. Mendapatkan hasil rekomendasi kata yang sebenarnya dengan menggunakan tingkat akurasi *Suggestion Adequacy (SA)*.

1.5 Manfaat Penelitian

Manfaat yang ingin dicapai dalam implementasi tugas akhir ini adalah :

1. Manfaat bagi pengguna dapat dijadikan acuan untuk pengejaan dalam penulisan Bahasa Jawa ngoko.
2. Manfaat bagi peneliti selanjutnya dapat dijadikan sebagai bahan informasi untuk penelitian selanjutnya.

1.6 Sistematika Penulisan

Sistematika penulisan dari tugas akhir ini ini terdiri dari lima bagian utama sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini berisi teori-teori yang digunakan pada penelitian ini. Teori-teori yang berhubungan dengan Bahasa Jawa ngoko dan metode *Levenshtein Distance*.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi tahapan–tahapan dalam pengumpulan data, perancangaan sistem perumusan masalah dan analisa.

BAB 4 ANALISA DAN PERANCANGAN

Bab ini berisi analisa dan perancangan aplikasi penerapan algoritma *Levenshtein Distance* dalam normalisasi kata Bahasa Jawa ngoko.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dari analisa dan perancangan dan pengujian pada aplikasi yang berhasil dibangun.

BAB 6 PENUTUP

Bab ini berisi rangkuman dari hasil penelitian yang telah dilakukan dan saran – saran untuk pengembangan aplikasi atau penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 *Artificial Intelligence* (AI)

AI adalah bidang studi yang berhubungan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia dalam sebuah sistem teknologi sehingga sistem tersebut dapat memfasilitasi proses pengambilan keputusan yang biasanya dilakukan oleh manusia. Pada dasarnya, kecerdasan buatan adalah suatu pengetahuan yang dapat membuat komputer memiliki kecerdasan layaknya manusia sehingga dapat melakukan hal-hal yang biasanya dikerjakan manusia dimana membutuhkan suatu kecerdasan tertentu; misalnya melakukan penalaran untuk mengambil suatu kesimpulan, memahami ucapan manusia, bersaing di level tertinggi pada sistem permainan strategis, menerjemahkan suatu bahasa ke bahasa lainnya, dll [5].

Artificial intelligence adalah sebuah rancangan program yang memungkinkan komputer melakukan suatu tugas atau mengambil keputusan dengan meniru suatu cara berpikir dan penalaran manusia. Cara kerja *artificial intelligence* adalah menerima *Input*, kemudian diproses dan kemudian mengeluarkan *output* berupa suatu keputusan [6].

2.2 *Natural Language Processing* (NLP)

Natural Language Processing (NLP) merupakan salah satu cabang ilmu *Artificial Intelligence* (AI) yang berfokus pada pengolahan bahasa natural atau bahasa yang secara umum digunakan oleh manusia untuk berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh waktu agar dapat dipahami

terlebih dahulu agar selaras dengan yang dimaksudkan oleh *user*. Terdapat berbagai implementasi dari NLP, Diantaranya adalah chatbot (aplikasi chat dimana user seolah-olah dapat berkomunikasi dengan komputer), *Stamming* atau *Lemmatization* (pemotongan kata dalam bahasa tertentu sehingga menjadi bentuk dasar pengenalan fungsi setiap kata dan kalimat), *Summarization* (ringkasan dari bacaan), *Translation Tools* (menerjemahkan bahasa) dan aplikasi-aplikasi lain yang memungkinkan komputer mampu memahami instruksi bahasa yang diinputkan oleh *user* [7].

2.3 *Text Preprocessing*

Text preprocessing merupakan sebuah proses yang dilakukan untuk mempersiapkan data sebelum dilakukan proses yang lainnya. Persiapan data dilakukan dengan cara mengeliminasi data yang tidak sesuai atau mengubah data tersebut menjadi bentuk yang lebih mudah diproses oleh sistem [8].

Proses *preprocessing* ini meliputi beberapa proses yaitu :

1. *Case Folding*

   Case folding merupukan sebuah proses yang mengubah huruf yang terdapat pada sebuah kata dan proses penyeragaman bentuk huruf, dalam hal ini hanya menerima huruf a sampai z.

2. *Cleaning*

   Cleaning merupakan proses dalam melakukan pembersihan dari atribut-atribut yang tidak dibutuhkan dengan informasi yang ada pada datauntuk mengurangi noise pada proses klasifikasi.

3. Tokenizing

Tokenizing merupakan tahap untuk memisahkan kata perkata pada kalimat yang telah dilakukan proses *cleaning* dan *case folding*. *Tokenizing* dilakukan dengan memisahkan kalimat menjadi satuan kata dan yang menjadi pemisah dari setiap kata adalah karakter spasi.

4. Stemming

Stemming merupakan tahapan penghapusan kata imbuhan, sisipan dan akhiran. Tahapan ini dilakuka proses pemotongan imbuhan-imbuhan, sisipan, dan akhiran yang terdapat pada kata yang telah ditokenisasi.

2.4 *Text Mining*

Text mining dapat diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, secara otomatis mengekstrak informasi dari sumbersumber yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber [9].

Text mining adalah proses menambang data berupa teks dengan sumber data biasanya dari dokumen dan tujuannya adalah mencari kata - kata yang mewakili dalam dokumen sehingga dapat dilakukan analisa keterhubungan dalam dokumen. Data teks akan diproses menjadi data numerik agar dapat dilakukan proses lebih lanjut. Sehingga dalam *text mining* ada istilah *preprocessing data*, yaitu proses pendahulu yang diterapkan terhadap data teks yang bertujuan untuk menghasilkan data numeric [10].

2.4.1 Tahapan *Text Mining*

Secara umum, tahapan besar dalam *text mining* terdiri dari tiga bagian utama yakni *text preprocessing*, *feature selection,* dan *text analytic*. Penjelasan lebih lanjut dari tahap-tahap tersebut adalah sebagai berikut [11] :

1. *Text Preprocessing.*

   Tahapan ini adalah tahapan yang berfungsi untuk membersihkan teks sebelum diolah lebih lanjut. Data teks mentah yang tidak terstruktur memiliki cukup banyak *noise* seperti tanda baca, angka, imbuhan, karakter-karakter khusus, *slang word* dan lain sebagainya. Dalam tahapan ini, data teks tersebut dibersihkan sehingga tersisa bentuk dasarnya saja untuk keperluan analisis teks lebih lanjut.

2. *Feature Selection.*

   Tahapan ini berperan dalam menentukan *term*/kata kunci yang menjadi ciri dari suatu dokumen yang membedakan dokumen tersebut dengan dokumen yang lain dalam satu korpus. Dalam *text mining*, feature selection merupakan tahapan yang paling penting yang memiliki peran yang sangat signifikan dalam akurasi *text analytic*. Empat pendekatan yang paling umum digunakan dalam *feature selection* adalah *Document Frequency* (DF), *Term Frequency* (TF), *Inverse Document Frequency* (IDF) dan *Term Frequency/Inverse Document Frequency* (TF/IDF).

   a. *Document Frequency* (DF).

      Prinsip kerja dari DF adalah membuang *term-term* yang umum terdapat di dokumen-dokumen yang ada pada suatu korpus dokumen teks.

Sehingga *term* yang tersisa dalam suatu dokumen adalah *term-term* yang memiliki tingkat *overlapping* yang rendah dengan *term-term* yang terdapat di dokumen lain dalam suatu korpus.

b. *Term Frequency* (TF).

Berbeda dengan DF, pendekatan TF tidak mengindahkan *term* yang terkandung dalam dokumen lain. Metode TF hanya secara sederhana menghitung kemunculan *term* dalam suatu dokumen. *Term-term* yang memiliki frekuensi kemunculan tinggi akan menjadi ciri dari suatu dokumen dimana *term* tersebut berada.

c. *Inverse Document Frequency* (IDF).

Pendekatan IDF mirip dengan TF, yakni menghitung frekuensi kemunculan suatu term. Namun, jika TF menghitung kemunculan suatu *term* hanya di satu dokumen teks, maka IDF menghitung kemunculan suatu term di keseluruhan korpus dokumen.

d. *Term Frequency/Inverse Document Frequency* (TF/IDF).

TF/IDF adalah gabungan dari pendekatan TF dan IDF dengan mengambil rasio antara nilai TF dan nilai IDF.

3. *Text Analytic.*

Tahapan terakhir dari proses *text mining* adalah *text analytic*. Dalam tahapan ini data teks yang sudah dibersihkan dan diidentifikasikan berdasarkan *term*/kata kunci yang menjadi ciri dokumen teks tersebut diolah dengan menggunakan berbagai macam algoritma untuk berbagai kebutuhan analisis. Dua jenis *text analytic* yang paling sering dilakukan

adalah *topic modelling* dan *sentiment analysis*. *Topic modelling* adalah sebuah pendekatan untuk mengelompokkan teks/dokumen teks kedalam beberapa kategori secara otomatis berdasarkan tingkat kesamaan *term*/kata kunci.

2.5 Normalisasi Teks

Normalisasi teks adalah mengubah teks kalimat menjadi teks yang secara lengkap memperlihatkan cara pengucapannya. Normalisasi teks meliputi pengubahan singkatan, akronim, angka, tanggal, waktu, karakter-karakter khusus, dan simbol-simbol dengan bentuk huruf alphabet lengkap sehingga tidak terjadi ambiguitas berkenaan dengan cara pengucapan. Terdapat beberapa pendekatan metode yang digunakan untuk melakukan normalisasi teks yaitu, pendekatan *Dictionary-Based, Lexical-Approach dan Machine Learning Based* [12].

a. *Dictionary-Based*

   Dictionary-Based adalah salah satu metode yang digunakan untuk menyelesaikan normalisasi teks. Proses normalisasi akan menggunakan berbagai macam model kamus kata sebagai acuan. Ada beberapa contoh kamus kata seperti kamus kata dasar, kamus singkatan, kamus kata slang, kamus istilah, dan lain sebagainya.

b. *Lexical-approach*

   Lexical-approach adalah metode dalam normalisasi teks yang menggunakan pola susunan kata yang diatur dalam *Rule-Based* terbentuk sebagai acuan dalam proses normalisasi. Pada metode ini, kata yang akan

dinormalisasi akan dideteksi berdasarkan jenis kata yang terdapat pada *Rule-Based* tersebut.

c. *Machine Learning Approach*

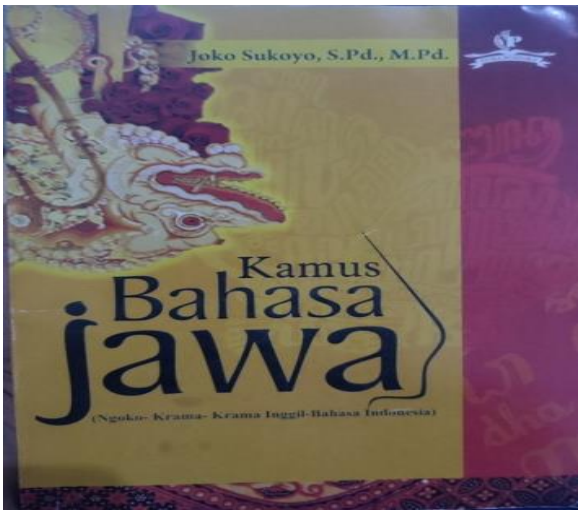
Machine learning approach adalah metode normalisasi yang menggunakan model *machine learning* dalam memeproses kata yang akan dinormalisasi. Model akan dilatih menggunakan dataset dari berbagai kata agar nantinya dapat mengenali bentuk kata yang akan dinormalisasi sehingga menghasilkan bentuk kata yang sesuai.

2.6 Bahasa Jawa

Bahasa Jawa merupakan salah satu bahasa daerah di Indonesia yang jumlah pemakaiannya cukup besar, yaitu sekitar 50% dari seluruh penduduk Indonesia. Berbagai bahasa Jawa tersebut masing-masing mempunyai variasi bahasa yang khas dan perbedaan variasi bahasa disebut dialek. Secara geografis, bahasa Jawa merupakan bahasa yang dipakai di wilayah provinsi Jawa Tengah, Daerah Istimewa Yogyakarta, dan Jawa Timur. Bahasa Jawa juga dituturkan oleh masyarakat Jawa yang bertransmigrasi ke luar Jawa seperti di Sumatera, kalimantan, Riau, dan beberapa provinsi lainnya di Indonesia. Bahasa Jawa merupakan bahasa yang mengenal adanya tingkat tutur atau undha-usuk basa atau unggah unggah basa. Adanya tingkat tutur dalam bahasa Jawa merupakan adat sopan santun berbahasa Jawa [13].

2.6.1 Bahasa Jawa Ngoko

Menurut Harjawinaya dan Supriya (2001:17-19) undha-usuk basa dipilah menjadi dua, yaitu undha usuk basa di zaman kejawen dan undha usuk basa di zaman modern. Undha-usuk basa Jawa di zaman *kejawen* mengenal enam tingkat tutur. Sedangkan Undha-usuk di zaman modern mengenal dua tingkat tutur (Harjawiyana dan Supriya, 2001:18). Tingkat tutur tersebut adalah zaman kejawen : Basa ngoko, Basa madya, Basa krama desa, Basa krama, Basa krama inggil, dan Basa kedhaton. zaman modern : Basa ngoko dan Basa krama [14]



Gambar 2.1 Kamus Bahasa Jawa

Tabel 2.1 Kosa Kata Jawa Ngoko

No	Kosa Kata Ngoko	Arti
1	Abang	Merah
2	Rai	Muka
3	Duwur	Atas
4	Sesuk	Besok
5	Kabeh	Semua

2.7 *Levenshtein Distanc*e

Levenshtein Distance atau *Edit Distance* adalah matriks perbandingan untuk mengukur perbedaan diantara dua urutan oleh Vladimir Levenshtein (1966). *Levenshtein Distance* sering dipakai dalam membandingkan antara dua urutan String yang berguna untuk masalah memperbaiki kesalahan eja dalam kata. Secara rumus matematika *Levenshtein Distance* bisa dinotasikan seperti persamaan 1 [15].

$$D_{(s,t)}= \begin{cases} \min D(s-1,t)+1 \text{ (penghapusan)} \\ \min D(s,t-1)+1 \text{(penyisipan)} \\ \min D(s-1,\ t-1)+1(a_i \neq b_j) \text{(Penukaran)} \\ \min D(s-1,t-1) s_j=t_i \text{(Tidak Ada Perubahan)} \end{cases} \quad \text{.....(1)}$$

Keterangan :

s = *String* sumber

t = *String* target

d = jarak *Levenshtein distance*

s_j = Karakter string member ke –j

t_i = Karakter string target ke –i

Ada tiga macam operasi yang digunakan oleh algoritma ini yaitu [16] :

1. Operasi Penggantian (*Substitution*)

   Operasi ini menukar karakter terhadap kata yang diindikasikan terdapat kesalahan eja.

2. Operasi Penambahan (*Insertion*)

   Operasi ini menambah karakter pada huruf pada kata yang diindikasikan terdapat kesalahan eja.

3. Operasi Penghapusan (*Deletion*)

   Operasi ini menghapus karakter pada huruf pada kata yang diindikasikan terdapat kesalahan eja.

2.8 Suggestion Adequacy (SA)

Suggestion adequacy **(***SA***)** merujuk pada kemampuan pemeriksa ejaan untuk menyajikan saran yang relevan dan akurat kepada pengguna untuk semua *true negatives* (yaitu kata-kata yang ditandai oleh pengejaan). Perhatikan bahwa SA dari pengejaan hanya harus didasarkan pada *true negatives* dan bukan pada *all negatives*, karena tujuannya adalah untuk menentukan seberapa baik pengeja bisa menyarankan rekomendasi untuk kata-kata yang salah.

Dengan sistem penilaian sebagai berikut:

- Rekomendasi kata yang tepat muncul pertama = nilai 1
- Rekomendasi kata yang tepat muncul kedua dan seterusnya = nilai 0.5
- Rekomendasi kata yang muncul tidak sama dengan hasil manusia = nilai -0.5
- Tidak ada saran = 0 (NS)

Oleh karena itu, untuk setiap rekomendasi yang benar dan muncul pertama, pemeriksa ejaan memberi nilai 1, dan untuk setiap rekomendasi benar muncul kedua dan seterusnya bernilai 0,5, jika pemeriksa ejaan hanya menawarkan rekomendasi yang salah akan diberikan nilai -0.5. Namun, jika pemeriksa ejaan tidak menawarkan rekomendasi nilainya.

Rumus akurasi sebagai berikut :

$$SA = \frac{\sum_{K=0}^{n} Sk}{N_{Tn}}$$

Persamaan 2. 2 Rumus *Suggestion Adequacy*

2.9 Algoritma

Algoritma adalah kumpulan dari instruksi berurutan, jelas dan terperinci atau salah satu urutan langkah-langkah pendekatan yang dilakukan seseorang untuk memecahkan sebuah persoalan yang sedang dihadapi, membagi masalah menjadi masalah yang lebih kecil (sub masalah) sehingga dapat dipecahkan dengan baik. Sebuah algoritma dibentuk oleh urutan proses yang benar dengan memperhatikan keadaan awal dan keadaan akhir dari sebuah persoalan [17].

2.10 Basis Data

Basis data adalah kumpulan *file-file* yang saling berelasi, relasi tersebut biasa ditunjukan dengan kunci dari tiap *file* yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu file terdapat *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk menunjukan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. Suatu sistem manajemen basis data berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi sistem manajemen basis data dan set program pengelola untuk menambah data, menghapus data, mengambil data dan membaca data [18].

2.11 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. *Relational Database Management System* (RDBMS). MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*) [19].

2.12 PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasiskan kode – kode *(script)* yang digunakan

untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML" [20].

2.13 Flowchart

Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian [21].

2.14 Context Diagram

Context Diagram adalah gambaran umum tentang suatu sistem yang terdapat didalam suatu organisasi yang memperlihatkan batasan (*boundary*) sistem, adanya interaksi antara eksternal *entity* dengan suatu sistem dan informasi secara umum mengalir di antara *entity* dan sistem. *Context Diagram* merupakan alat bantu yang digunakan dalam menganalisa sistem yang akan dikembangkan. Simbol-simbol yang digunakan di dalam *Context Diagram* hampir sama dengan simbol-simbol yang ada pada DFD, hanya saja pada *Context Diagram* tidak terdapat simbol file [22].

2.15 Data Flow Diagram (DFD)

Alat utama untuk merepresentasikan proses komponen sistem dan arus data di antaranya adalah *data flow diagram* (DFD). *Data flow diagram* menawarkan model grafis logis dari arus informasi, mempartisi sistem menjadi modul yang menunjukkan tingkat detail yang dapat diatur. Ini secara ketat

menentukan proses atau transformasi yang terjadi di dalam setiap modul dan antarmuka yang ada di antara keduanya. Adapun keuntungan menggunakan DFD adalah meningkatkan pemahaman keterkaitan antara sistem dan sub sistem, selain sebagai alat yang efektif dalam berkomunikasi dengan pengguna [23].

2.16 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah jenis model basis data berdasarkan pengertian suatu entitas dunia nyata dan hubungan di antara mereka. Kita dapat memetakan skenario dunia nyata ke model database hubungan antar entitas. Model hubungan entitas ini menciptakan satu set entitas dengan atributnya, satu set konstrain dan relasi di antara keduanya [24].

2.17 Penelitian Terdahulu

Adapun penelitian terdahulu tentang metode *Levenshtein Distance* dapat dilihat pada tabel 2.1 berikut :

Tabel 2.2 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
1	Rachmania Nur Dwitiyastuti, Adharul Muttaqin dan Muhammad Aswin	Pengoreksi Kesalahan ejaan bahasa indonesia menggunakan metode *Levenshtein Distance*	Berdasarkan hasil pengujian pada sistem pengoreksi kesalahan ejaan, sistem sudah dapat memeriksa keberadaan kata pada kamus dan memberikan sugesti untuk kata yang salah dengan menggunakan metode *Levenshtein Distance.* Berdasarkan hasil pengujian pencarian kata, sistem bisa mendeteksi 100% kesalahan kata. Dari hasil pengujian pencarian sugesti

			didapatkan bahwa sistem sudah bisa mendeteksi kesalahan dan memberi sugesti untuk tiap kategori kesalahan ejaan.
2	Na'firul Hasna Ariyani, Sutardi, Rahmat Ramadhan, 2016	Aplikasi pendeteksi kemiripan isi teks dokumen menggunakan metode *Levenshtein Distance*	Pada pengujian menggunakan data real yaitu data dokumen berplagiat yang diambil dari artikel/berita lewat internet, algoritma *Levenshtein Distance* menghasilkan nilai *similarity* yang tinggi yaitu diatas 85 % sampai 100 % untuk dokumen yang tingkat kemiripannya tinggi. Sedangkan untuk dokumen dengan tingkat kemiripan yang rendah atau tidak berplagiat maka menghasilkan nilai similarity dibawah 40%.
3	Dewi Rosmala, Zulfikar Muhammad Risyad	Algoritma Levenshtein distance dalam aplikasi pencarian kata isu di kota Bandung pada Twitter	Dari hasil implementasi dan pengujian fungsi yang telah dilakukan, dapat disimpulkan bahwa penggunaan Algoritma Levenshtein Distance dalam Aplikasi Pencarian Kata Isu di Kota Bandung Pada Twitter mampu mengubah tweet pelaporan isu yang ditujukan pada akun Twitter infobdg dan dinas-dinas Pemerintah Kota Bandung yang mengandung kata dengan kesalahan ejaan menjadi kata kunci yang kemudian dimasukkan ke dalam daftar kategori isu yang terdapat di Pemerintah Kota Bandung dengan tingkat akurasi 100%. Dengan digunakannya

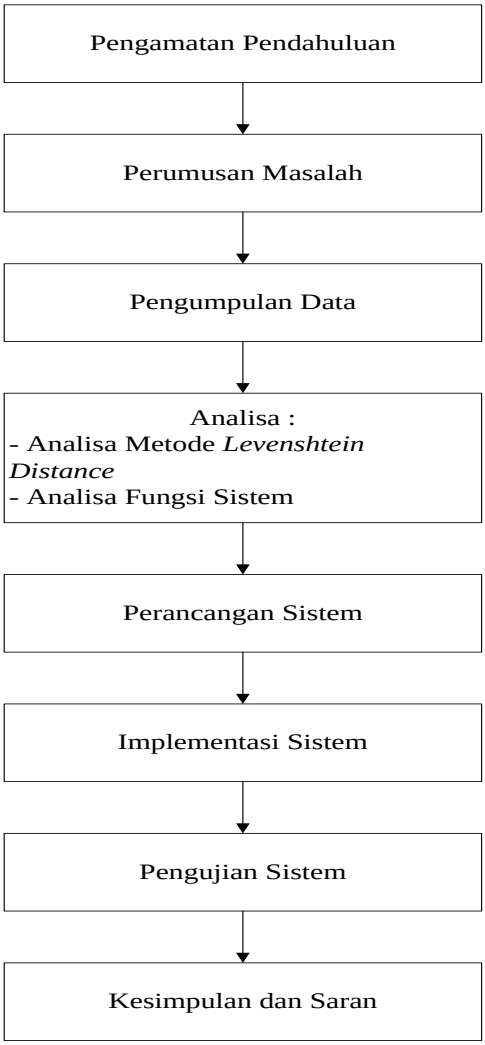
			Algoritma Levenshtein Distance dapat memberikan akurasi data yang lebih baik pada hasil keluaran aplikasi, sehingga dapat digunakan Pemerintah Kota Bandung dalam memperoleh data isu-isu yang dilaporkan warga kepada akun Twitter infobdg dan dinas Pemerintah Kota Bandung.
4	Stezar Priyansyah & Renny Pradina	Normalisasi Teks Media Sosial Menggunakan Word2vec, Levenshtein Distance, dan Jaro-Winkler Distance	Penelitian mendapatkan akurasi 25% pelatihan model word2vec dan pengujian sampel terbaik pengujian data mendapatkan akurasi terbaik adalah 75.9% dengan threshold sebesar 70%.
5	Ahmad Khairun Arsyad, Bambang Pramono, Isnawaty, Muhammad Yamin, Ihsan	Implementasi *Levenshtein Distance* Pada Aplikasi Pencarian Barang Di Berbagai *E-Marketplace* Menggunakan Teknik *Web Scraping*	Adapun beberapa saran untuk pengembangan lebih lanjut dari Implementasi *Levenshtein Distance* pada Aplikasi Pencarian Barang di Berbagai *E-Marketplace* Menggunakan Teknik *Web Scraping*, yaitu: 1. Sistem diharapkan dapat dikembangkan lagi dengan menambahkan jumlah *e-marketplace* yang terdaftar serta penambahan fitur seperti filter kategori barang terlaris dan filter kategori barang diskon. 2. Sistem diharapkan mampu menangani *error* yang terjadi dalam proses *web scraping* dengan lebih cerdas terutama penanganan *error* jika terjadi perubahan tampilan dari *website* yang di-*scraping*

6	Arrofi Reza Satria, sigit Adinugroho, suprapto	Analisis Sentimen Ulasan Aplikasi *Mobile* menggunakan Algoritma Gabungan Naïve Bayes dan C4.5 berbasis Normalisasi Kata Levenshtein Distance	Berdasarkan analisa dari penggunaan metode Levenshtein Distance dan Naïve Bayes – C4.5 dalam klasifikasi sentiment aplikasi mobile terdapat banyak faktor yang mempengaruhi nilai akurasi klasifikasi. Faktor pertama adalah perbedaan yang signifikan antara jumlah data sentimen positif dan negatif. Dalam hal ini mempengaruhi nilai waktu komputasi dan nilai akurasi dari masing masing sentimen tersebut. Faktor kedua adalah banyaknya kata yang terdapat salah eja, tidak baku, kata baru dan kata yang bukan dari bahasa Indonesia. Kata – kata ini akan menimbulkan kesalahan dalam proses klasifikasi. Faktor ketiga adalah penerapan Levenstein Distance yang dinilai tidak efektif secara signifikan mempengaruhi hasil klasifikasi. Secara keseluruhan penelitian penggunaan metode *Levenshtein Distance* dan Naïve Bayes – C4.5 dalam menganalisis sentimen aplikasi mobile memiliki nilai akurasi diatas 85.3% dengan nilai akurasi tertinggi sebesar 87.1%

BAB 3

METODOLOGI PENELITIAN

Tahapan penelitian yang akan dilakukan dan penyelesaian masalah terhadap normalisasi Bahasa Jawa Ngoko dengan menggunakan metode *Levenshtein Distance*. Adapun tahapan metodologi yang dilakukan selama penelitian dapat dilihat pada gambar 3.1, yang mana merupakan proses yang dimulai dari studi literatur hingga diperoleh kesimpulan.



Gambar 3.1 Tahapan Metodologi Penelitian

Pembuatan tugas akhir ini terbagi menjadi beberapa tahap pengerjaan yang tertera sebagai berikut :

3.1 Pengamatan Pendahuluan

Pengamatan pendahuluan merupakan tahapan awal yang dilakukan dalam penelitian ini, yang menggunakan metode *Levenshtein Distance* yang dijadikan sebagai penelitian studi pustaka dalam penelitian Tugas Akhir ini. Pada penelitian metode ini yaitu menggunakan kosa kata Bahasa Jawa Ngoko.

3.2 Perumusan Masalah

Berdasarkan hasil dari tahapan pengamatan pendahuluan sebelumnya, maka tahapan selanjutnya adalah tahapan perumusan masalah. Pada tahapan perumusan masalah akan dirumuskan masalah yang dianggap sebagai penelitian dalam Tugas Akhir ini. Permasalahan-permasalahan yang menjadi rumusan masalah dalam penelitian ini didapatkan dari penelitian dari penelitian terkait data pengamatan pendahuluan sebelumnya. Solusi yang didapatkan pada tahapan perumusan masalah ini yang akan menjadi judul penelitian Tugas Akhir ini "Pengembangan Koreksi Ejaan Bahasa Jawa Ngoko menggunakan algoritma *Levenshtein Distance* dengan tingkat akurasi *Suggestion Adequacy*".

3.3 Pengumpulan Data

Pengumpulan data adalah tahapan-tahapan yang bertujuan dalam memperoleh data-data informasi yang berhubungan dengan penelitian Tugas Akhir ini. Pada tahapan pengumpulan data ini juga berguna untuk mengumpulkan semua kebutuhan data yang akan diproses nantinya menggunakan metode

"*Levenshtein Distance*". Dalam pengumpulan data ini data yang dikutip adalah sebagai berikut :

1. Studi Literatur

   Dalam proses penelitian, diperlukan pengumpulan pengetahuan dengan cara mempelajari literatur dari beberapa bidang ilmu yang berhubungan dengan normalisasi Bahasa Jawa Ngoko dengan menggunakan metode *levenshtein distance*, yaitu diantaranya:

   a. Pengumpulan informasi mengenai proses normalisasi teks.
   b. Pengumpulan informasi terkait metode *levenshtein distance*
   c. Pengumpulan data dari jurnal dan buku-buku.
   d. Pengumpulan informasi tentang penelitian terkait.

2. Wawancara

   Wawancara dilakukan untuk mendapatkan data penelitian. Dalam hal ini wawancara dilakukan kepada tokoh adat yang paham tentang Bahasa Jawa Ngoko di Desa Bukit Senyum.

3. Analisis Kebutuhan Sistem

   Dalam analisa sistem bertujuan mengidentifikasi sistem yang akan dirancang, yang meliputi perangkat lunak serta perangkat keras. Tahapan-tahapan yang menyusun analisa sistem ini adalah analisis data yang dipakai, spesifikasi kebutuhan sistem, spesifikasi pengguna, perancangan basis data, dan perancangan antarmuka. Adapun spesifikasi dari perangkat keras (*hardware*) dan perangkat lunak software (*software*) yang digunakan sebagai berikut :

a. Perangkat keras (*hardware*), antara lain :

   Prosesor : Intel(R) Core(TM) i5-650

   *Memory (*RAM) : 4.00 GB

   System type : 64-bit *Operating system*

   Harddisk : 500 GB

b. Perangkat Lunak (*software*), antara lain :

   Sistem Operasi : *Windows* 7

   Tool : Microsoft Office, Xampp, Notepad ++, *Google Chrome*

3.4 Analisa Sistem

Tahapan selanjutnya adalah melakukan analisis metode sistem penelitian tugas akhir ini. Adapun tahapan analisa dalam penelitian ini sebagai berikut :

3.4.1 Analisa Metode *Levenshtein Distance*

Tahapan ini adalah proses dimana langkah-langkah pengolahan data menggunakan metode *Levenshtein Distance* dijalankan.

3.4.2 Analisa Fungsi Sistem

Setelah melakukan tahapan analisis terhadap metode *Levenshtein Distance* selanjutnya adalah analisis fungsional yang akan dibangun. Adapaun tahapan-tahapan analisis fungsional yaitu dalam pembuatan *flowchart*, *context diagram*, *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD) dan perancangan *user interface*.

3.5 Perancangan Sistem

Setelah tahapan analisis dilakukan, maka tahapan selanjutnya adalah perancangan sistem. Tahapan perancangan sistem terdiri dari :

1. Perancangan struktur menu yang akan digunakan pada sistem yang akan dibangun.
2. Tahapan rancangan *database* beserta atribut yang dibutuhkan.
3. Tahapan perancangan *user interface* atau antar muka pengguna terhadap sistem yang akan digunakan

3.6 Implementasi Sistem

Implementasi sistem merupakan suatu konversi dari desain aplikasi yang telah dirancang ke dalam sebuah program komputer dengan aplikasi berbasis *web* dengan menggunakan bahasa pemrograman HTML, PHP, CSS dan JavaScript serta penyimpanan database yang menggunakan MySQL.

3.7 Pengujian Sistem

Pengujian merupakan sebuah tahapan yang memperlihatkan apakah prediksi tingkat akurasi dari penelitian sesuai dengan yang diinginkan atau tidak. Pengujian yang dilakukan terdiri dari :

1. Pengujian *black box,* digunakan untuk menguji tingkat kemampuan *user interface* terhadap sistem yang dibangun.
2. Pengujian *User Acceptance Test* (UAT).

3.8 Kesimpulan dan Saran

Tahapan terakhir adalah menarik kesimpulan dari hasil penelitian yang didapatkan dalam Pengembangan Koreksi Ejaan Bahasa Jawa Ngoko

menggunakan algoritma *Levenshtein Distance* dengan tingkat akurasi *Suggestion Adequacy*. Pada tahapan ini juga berisikan saran peneliti bagi pembaca untuk melakukan pengembangan terhadap penelitian ini kedepannya.