

**NORMALISASI MIKROTEKS BERBENTUK SINGKATAN PADA TEKS
TWITTER BERBAHASA INDONESIA MENGGUNAKAN ALGORITMA
*LONGEST COMMON SUBSEQUENCES***

SKRIPSI

ZURWATUS SANIYAH

141402005



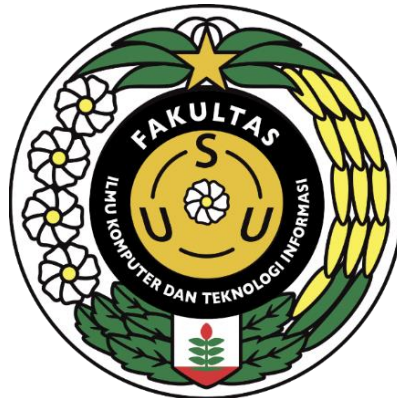
**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2019**

**MICROTEXT NORMALIZATION THE FORM OF ABBREVIATIONS IN
INDONESIAN TWITTER TEXT USING THE LONGEST COMMON
SUBSEQUENCES ALGORITHM**

SKRIPSI

ZURWATUS SANIYAH

141402005



PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2019

**NORMALISASI MIKROTEKS BERBENTUK SINGKATAN PADA TEKS
TWITTER BERBAHASA INDONESIA MENGGUNAKAN ALGORITMA
*LONGEST COMMON SUBSEQUENCES***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Teknologi Informasi

ZURWATUS SANIYAH

141402005



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2019**

PERSETUJUAN

Judul : NORMALISASI MIKROTEKS BERBENTUK
SINGKATAN PADA TEKS TWITTER
BERBAHASA INDONESIA MENGGUNAKAN
ALGORITMA *LONGEST COMMON*
SUBSEQUENCES

Kategori : SKRIPSI

Nama : ZURWATUS SANIYAH

Nomor Induk Mahasiswa : 141402005

Program Studi : S1 TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI
INFORMASI

UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :

Pembimbing 2



Ainul Hizriadi, S.Kom, M.Sc
NIP. 19851027 201706 1 001

Pembimbing 1



Dani Gunawan, ST., M.T.
NIP. 19820915 201212 1 002

Diketahui/disetujui oleh

Program Studi S1 Teknologi Informasi

Ketua,



Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
NIP. 19860303 201012 1 004

PERNYATAAN

NORMALISASI MIKROTEKS BERBENTUK SINGKATAN PADA TEKS
TWITTER BERBAHASA INDONESIA MENGGUNAKAN ALGORITMA
LONGEST COMMON SUBSEQUENCES

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 25 April 2019



Zurwatus Saniyah

141402005

UCAPAN TERIMA KASIH

Alhamdulillah, puji dan syukur penulis sampaikan kehadiran Allah SWT yang telah memberikan rahmat dan izin-Nya sehingga penulis dapat menyelesaikan skripsi ini sebagai syarat untuk memperoleh gelar sarjana Komputer pada Program Studi S1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Selama pengerjaan tugas akhir ini, banyak sekali bantuan dan dukungan serta doa dari berbagai pihak sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih sedalam-dalamnya dan penghargaan kepada:

1. Kedua orang tua penulis, Ayahanda Syahlan, S.Pd dan Ibunda Rida Nursiama, S.Pd, yang tidak pernah berhenti memberikan dukungan serta doanya kepada penulis. Orangtua penulis selama ini tidak pernah memberikan tekanan dalam hal apapun kepada penulis dan selalu mengerti keadaan penulis, sehingga menjadi motivator terbesar penulis dalam menjalani masa perkuliahan serta penulis dapat menyelesaikan tugas akhir dengan baik. Ucapan terima kasih juga kepada adik penulis, Muhammad Fahmi yang telah memberikan dukungan dan doa kepada penulis.
2. Bapak Dani Gunawan, ST., MT. selaku dosen pembimbing pertama dan kepada Bapak Ainul Hizriadi, S.Kom, M.Sc. selaku dosen pembimbing kedua yang telah bersedia meluangkan waktu dan pikirannya untuk membimbing penulis dalam menyelesaikan tugas akhir ini, baik dalam pengerjaan program maupun penulisan skripsi.
3. Penulis juga mengucapkan terima kasih kepada Bapak Romi Fadillah Rahmat, B.Comp.Sc., M.Sc. selaku dosen pembimbing pertama dan Bapak Dr. Sawaluddin, M.IT selaku dosen pembimbing kedua yang telah memberikan kritik dan saran pada hasil penelitian penulis dan penulisan skripsi penulis.
4. Penulis mengucapkan terima kasih kepada dosen, pegawai, dan staff di lingkungan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang telah membantu dalam penyelesaian administrasi dan membimbing penulis selama masa perkuliahan.

5. Ucapan terima kasih kepada Aggie Wicita Rini Riady dan Afzalurrahmah, sahabat sekaligus teman seperjuangan dibawah bimbingan Bapak Dani Gunawan, ST., MT yang banyak memberikan motivasi dan dukungan dalam proses pengerjaan skripsi.
6. Penulis mengucapkan terima kasih kepada sahabat penulis yang sudah berteman dekat semasa kuliah, Sigit Satrio, Aggie Wicita Rini Riady, Ayu Astuti Kartika, Fanny Ramadhana, Nadya Maysyarah, Novy Yolanda, Cindy Pakpahan, Afzalurrahmah, Sity Ayu Novarina Suyanto, Varuna Dewi, Hetly Saint Kartika, Victoria Tambunan, sahabat yang selalu berbagi pengalaman dan menjadi tempat bercerita semasa perkuliahan, serta selalu memberikan kritik dan saran dalam proses pengerjaan skripsi. Sahabat yang sudah seperti keluarga sendiri yang saling mendukung dan memberikan motivasi semasa perkuliahan hingga sekarang.
7. Ucapan terima kasih kepada Muhammad Fachrin Aulia Nasution, Muhammad Fadly Tanjung, Syaiful Anwar Husein Lubis, Rendra Mahardika, Muhammad Noor Misyuari, Mahabatan, yang telah membagikan waktu dan ilmunya untuk membantu penulis dalam proses pengerjaan skripsi.
8. Ucapan terima kasih kepada teman dan sahabat penulis sejak masa sekolah, terkhusus kepada Putri Ramadhani, Ade Tria Novyanti, Asry Kartika Dwy, Swedio Fransteddy, Muhammad Meyra Trisna, Rusdiansyah, yang sampai sekarang selalu memberikan semangat, dukungan, dan doanya kepada penulis.
9. Ucapan terima kasih kepada seluruh teman angkatan Teknologi Informasi 2014 yang sudah berteman baik selama ini dan saling berjuang bersama dalam mengerjakan skripsi terkhusus kepada Sry Anggraini, Yunda Andriyani, Muhammad Faris Pratama, Muhammad Aidiel Rachman, Muhammad Anggi Lianda, Sakta Akbari, Duwi Satria Kurniawan, Ibnu Habibie, Muhammad Abror Rambe, Ridho Fariha, Isa Dadi, yang telah menjadi teman belajar, bercerita, dan bermain selama masa kuliah.
10. Penulis mengucapkan terima kasih kepada HIMATIF USU yang telah menjadi tempat bagi penulis dalam belajar berorganisasi.
11. Terima kasih juga penulis ucapkan untuk semua pihak yang telah terlibat dalam masa perkuliahan dan pengerjaan tugas akhir, sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.

Semoga Allah SWT melimpahkan berkah, rahmat, dan karunia-Nya kepada semua pihak yang telah memberikan bantuan, perhatian dan dukungan kepada penulis dalam menyelesaikan skripsi ini.

Medan, 25 April 2019

A handwritten signature in black ink, consisting of stylized cursive letters, likely representing the author's name.

Penulis

ABSTRAK

Media sosial merupakan perkembangan teknologi yang saat ini menjadi kebutuhan utama bagi setiap orang sebagai pemberi dan penerima informasi. Twitter merupakan salah satu media sosial dan *microblog* yang paling populer untuk berkomunikasi dan menyampaikan informasi. Namun, Twitter membatasi ruang penulisan karakter atau dikenal dengan istilah mikroteks. Mikroteks adalah batasan ruang penulisan dalam cakupan kecil. Keterbatasan karakter tersebut menyebabkan pengguna Twitter melakukan penyingkatan kata untuk memaksimalkan informasi yang disampaikan, sehingga membawa dampak terhadap kualitas teks yang dihasilkan. Diperlukan pemrosesan otomatis untuk melakukan proses normalisasi. Normalisasi yang dilakukan dalam penelitian ini yaitu mengubah kata tidak baku menjadi kata baku sesuai dengan kamus Bahasa Indonesia menggunakan *dictionary based* dan algoritma *Longest Common Subsequences*. Proses normalisasi menggunakan *dictionary based* akan membaca *string* kata masukan dan membandingkannya dengan isi kamus singkatan. Jika *output* tidak ditemukan, proses normalisasi akan dilanjutkan menggunakan algoritma LCS. Algoritma LCS akan mencari kedekatan kata yang memiliki *string* yang sama dengan masukan yaitu dengan menggunakan perhitungan matriks. Tingkat akurasi yang dihasilkan sebesar 83%, presisi 90%, recall 87%, dan F1-Score 0.88 dengan jumlah data uji sebanyak 400 *tweet*.

Kata kunci : *Longest Common Subsequences*, Mikroteks, Normalisasi Teks, Twitter

MICROTEXT NORMALIZATION THE FORM OF ABBREVIATIONS IN INDONESIAN TWITTER TEXT USING THE LONGEST COMMON SUBSEQUENCES ALGORITHM

ABSTRACT

Social media is a technological development and nowadays it is become main needs to everyone for giver and recipient of information. Twitter is one of the most popular social media and microblog for communicating and sharing information. But, Twitter limits the character writing space it called by microtext . Microtext is limited of writing space in small coverage. The limitation causes twitter users to abbreviate words for maximize the information to share, thus having an impact on the quality of the text produced. Automatic processing is needed to carry out the normalization process. Normalization carried out in this study is changing the non standard words according to Indonesian dictionary using dictionary based and Longest Common Subsequences algorithm. The process of normalizing using dictionary based will read the string of input words and compare with words contained in the abbreviation dictionary. If the output is not found, normalization processing will continue using LCS Algorithm. LCS algorithm is looking for similar words that have the same string with the input and it is using matrix calculation. The level of accuracy produced is 83 %, precision 90 %, recall 87 %, and F1- Score 0,88 with the amount of test data as many as 400 tweets.

Keywords : Longest Common Subsequences, Microtext, Normalization, Twitter

DAFTAR ISI

	Hal.
PERSETUJUAN	Error! Bookmark not defined.
PERNYATAAN	ii
UCAPAN TERIMA KASIH	iii
ABSTRAK	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Tujuan Penelitian	4
1.4. Batasan Masalah	4
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	5
1.7. Sistematika Penulisan	6
BAB 2	7
LANDASAN TEORI	7
2.1. Mikroteks	7
2.2. Normalisasi Teks	7
2.3. <i>Dictionary Based</i>	8
2.4. Twitter	9
2.5. <i>Natural Language Processing</i>	10
2.6. <i>Longest Common Subsequences</i>	12

2.7. Penelitian Terdahulu	15
BAB 3	19
ANALISIS DAN PERANCANGAN	19
3.1. Data yang Digunakan	19
3.2. Perancangan Sistem	19
BAB 4	38
IMPLEMENTASI DAN PENGUJIAN	38
4.1. Implementasi Sistem	38
4.2. Diskusi Proses Normalisasi	43
4.3. Pengujian Sistem	49
BAB 5	58
KESIMPULAN DAN SARAN	58
5.1. Kesimpulan	58
5.2. Saran	58
DAFTAR PUSTAKA	59
LAMPIRAN	61

DAFTAR TABEL

Tabel 2.1. Hasil kalkulasi LCS dari FTKP dan FOTOKOPI	14
Tabel 2.2. Hasil untuk mendapatkan LCS dari FTKP dan FOTOKOPI	14
Tabel 2.3. Penelitian Terdahulu	17
Tabel 3.1. Pengecekan <i>stemming</i>	23
Tabel 3.2. <i>Dictionary based</i>	25
Tabel 3.3. Contoh Beberapa Kata Uji Hasil LCS	29
Tabel 3.4. <i>Word Choices</i>	31
Tabel 4.1. Hasil Proses Pengecekan perkata	45
Tabel 4.2. Perbandingan <i>Tweet</i>	49
Tabel 4.3. Pengujian Sistem	50
Tabel 4.4. Hasil Pengujian Sistem Menggunakan Algoritma LCS	51
Tabel 4.5. Hasil Pengujian Sistem Menggunakan <i>Dictionary Based</i>	54
Tabel 4.6. Hasil Pengujian Sistem Menggunakan LCS dan <i>Dictionary Based</i>	55
Tabel 4.7. Perbandingan Hasil Perhitungan Akurasi, Presisi, Recall, dan <i>F-Score</i>	57

DAFTAR GAMBAR

Gambar 3.1. Arsitektur Umum	20
Gambar 3.2. Diagram Aktivitas Sistem	33
Gambar 3.3. <i>Flowchart</i> Proses Normalisasi	34
Gambar 3.4. Rancangan halaman <i>upload file</i>	36
Gambar 3.5. Rancangan halaman normalisasi	37
Gambar 4.1. Tampilan <i>Upload File</i>	38
Gambar 4.2. Tampilan Hasil Normalisasi	38
Gambar 4.3. Tampilan Pilih <i>File</i>	39
Gambar 4.4. Tampilan <i>Input File</i>	39
Gambar 4.5. Tampilan <i>Process Tweets</i>	40
Gambar 4.6. Tampilan Hasil Normalisasi	40
Gambar 4.7. Tampilan <i>Word Choices</i>	41
Gambar 4.8. Tampilan <i>Button Save Result</i>	42
Gambar 4.9. Tampilan Direktori Penyimpanan	42
Gambar 4.10. <i>File</i> Tersimpan	43

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Indonesia merupakan salah satu negara dengan angka penggunaan internet terbesar di dunia. Menurut hasil survei APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) pada tahun 2017, penetrasi pengguna internet di Indonesia mencapai 143,26 juta. Alasan umum pengguna internet menggunakan internet adalah untuk mendapatkan pembaruan informasi. Media sosial adalah salah satu konten terbanyak yang diakses oleh pengguna internet Indonesia untuk mendapatkan pembaruan informasi. Media sosial merupakan salah satu perkembangan teknologi yang menghubungkan setiap orang melalui jaringan internet sebagai pemberi dan penerima informasi. Saat ini, media sosial di Indonesia dapat dianggap sebagai kebutuhan utama untuk pembaruan informasi dikarenakan maraknya informasi – informasi terbaru yang disebarkan melalui media seperti Facebook, Twitter, Instagram, dan lainnya. Salah satu media sosial yang paling populer untuk berkomunikasi adalah Twitter.

Twitter yang berada pada peringkat 5 dengan pengguna sebanyak 7,2 juta. Hal ini dibuktikan dengan adanya 110 juta *tweet* per hari dan jumlah pengguna lebih dari 200 juta (Sarwani dan Mahmudy 2015). Twitter merupakan media sosial dan *microblog* di internet yang banyak digunakan untuk menyampaikan informasi, mengutarakan pendapat, ataupun mengutarakan perasaan pengguna. Namun, Twitter memiliki keterbatasan karakter penulisan yaitu hanya 280 karakter sehingga menyebabkan pengguna Twitter sering melakukan penyingkatan. Singkatan tersebut mengakibatkan kata menjadi tidak baku (Wahyuningtyas 2016).

Penulisan kata baku yang sengaja dipersingkat, contohnya kata ‘yang’ dipersingkat menjadi ‘yg’, kata ‘fotokopi’ dipersingkat menjadi ‘ftkp’, kata berimbuhan seperti ‘mencintai’ dipersingkat menjadi ‘mncintai’, dan lain sebagainya yang kecenderungan

penulisan teks pada suatu *tweet*. Salah satu tujuan pengguna Twitter melakukan penyingkatan kata tersebut yaitu untuk memanfaatkan ruang penulisan bagi pengguna yang ingin mengutarakan informasi lebih dari 280 karakter. Ruang teks yang memiliki batasan penulisan dalam cakupan kecil dikenal dengan istilah mikroteks. Penulisan mikroteks biasanya tidak melihat struktur kalimat, penulisan dengan pengucapan yang salah, dan melakukan penyingkatan. Penulisan seperti ini akan membawa dampak terhadap kualitas teks yang dihasilkan. Hal lain yang menyebabkan faktor ketidakbakuan kata atau penyingkatan kata adalah pengguna Twitter yang memang terbiasa melakukan penyingkatan kata dalam penyampaianannya.

Dalam melakukan pemrosesan data teks yang tidak terstruktur ini, para peneliti menggunakan metode yang disebut dengan *Natural Language Processing* atau yang biasa disingkat NLP. NLP adalah sebuah metode pembentukan model komputasi bahasa sebagai bentuk interaksi antara manusia dan komputer dengan perantara bahasa alami. NLP berupaya untuk dapat memecahkan masalah untuk memahami bahasa alami manusia, dengan segala aturan gramatika dan semantiknya, serta mengubah bahasa tersebut menjadi representasi formal yang dapat diproses oleh komputer.

Sebelumnya telah dijelaskan pada penelitian yang dilakukan oleh Xue Zhenzhen, *et al.* (2011) ada empat faktor yang menjadi permasalahan dalam penulisan mikroteks yaitu faktor ortografi, fonetik, kontekstual, dan akronim. Penelitian tersebut melakukan normalisasi mikroteks untuk semua jenis penulisan dengan menggunakan pendekatan *Channel Models* pada masing-masing faktor. *Grapheme Channel* pada ortografi, *Phoneme Channel* pada fonetik, *Context Channel* pada kontekstual, dan *Acronym Channel* pada akronim.

Pada tahun 2015 penelitian dilakukan oleh Khoury Richard (2015) yaitu melakukan normalisasi mikroteks pada sosial media menggunakan algoritma Levenshtein *distance*. Penelitian ini melakukan normalisasi pada masalah fonetik berdasarkan struktur OOV dalam Bahasa Inggris dan juga melakukan normalisasi untuk singkatan.

Penelitian selanjutnya dilakukan oleh Irawan Johanes (2016) yaitu melakukan normalisasi pada teks Twitter dengan menggunakan *Noisy Channel Model*. *Noisy Channel Model* menjelaskan bahwa bentuk yang terlihat pada suatu kata yang terlihat

dapatlah menjadi sebuah bentuk yang terdistorsi dari bentuk aslinya. Hasil dari penelitian ini, sistem mampu membedakan beberapa variasi dari sebuah kata, namun tidak dapat membedakan kata yang harusnya tidak dinormalisasi dan tidak dapat memperbaiki singkatan.

Pada tahun 2017, Saragih Tri (2017) melakukan normalisasi pada teks Twitter Berbahasa Indonesia. Pada penelitian tersebut dilakukan perbandingan metode yang ada pada algoritma jarak *string* (*stringdist*) menggunakan bahasa pemrograman R. Ada 10 metode yang digunakan untuk menghitung jumlah perbedaan jarak antar *string* yang selanjutnya digunakan untuk pengubahan *string*.

Selanjutnya, masih pada tahun 2017 penelitian juga dilakukan oleh Hanafiah Novita, *et al.* (2017). Penelitian ini melakukan normalisasi teks pada Twitter untuk *complaint category*. Penelitian ini memanfaatkan data Twitter khususnya Bahasa Indonesia. Proses dibagi dalam tiga tahap, yaitu proses pembersihan, deteksi OOV, dan penggantian kata.

Berdasarkan latar belakang tersebut, penulis melakukan penelitian normalisasi teks pada Twitter menggunakan metode dan algoritma yang berbeda. Judul dari penelitian tersebut yaitu “Normalisasi Mikroteks Berbentuk Singkatan Pada Teks Twitter Berbahasa Indonesia Menggunakan Algoritma Longest Common Subsequences”. Diharapkan dengan menerapkan metode dan algoritma ini memperoleh hasil perbaikan kata lebih baik dan dengan nilai akurasi yang lebih baik.

1.2. Rumusan Masalah

Pengguna media sosial Twitter memiliki keterbatasan ruang dalam menyampaikan pesan atau informasi. Karena keterbatasan karakter, pengguna sangat memanfaatkan ketersediaan ruang dengan menuliskan kata baku yang sengaja dipersingkat dan menggunakan gaya bahasa modern. Hal seperti ini tidak akan dapat dipahami oleh sistem dan juga untuk peneliti selanjutnya sebagai keperluan *text mining*. Oleh karena itu diperlukan normalisasi mikroteks untuk memperbaiki singkatan tersebut menjadi kata yang sebenarnya dengan mengacu pada kamus Bahasa Indonesia.

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah memperbaiki kata dari *tweet* yang berbentuk singkatan menjadi bentuk kata baku yang sebenarnya sesuai kamus Bahasa Indonesia dengan menggunakan algoritma *Longest Common Subsequences*.

1.4. Batasan Masalah

Batasan masalah yang diberikan dalam penelitian ini adalah :

1. Data yang digunakan dalam penelitian ini adalah data *tweet* dari Twitter, tetapi tidak termasuk pada fitur *hashtag*, *retweet*, dan *mention*.
2. Normalisasi yang dilakukan hanya pada *tweet* berbahasa Indonesia
3. Normalisasi yang dilakukan hanya dalam bentuk huruf
4. Normalisasi tidak mempertimbangkan struktur antar kalimat

1.5. Manfaat Penelitian

Dengan melakukan normalisasi dan memperbaiki singkatan kata tidak baku pada Twitter menjadi kata baku diharapkan akan bermanfaat untuk penelitian lain yang ingin menggunakan data normalisasi Twitter dalam pengklasifikasian data atau keperluan *text mining*.

1.6. Metodologi Penelitian

Adapun tahapan-tahapan yang dilakukan pada penelitian ini, yaitu :

1. Studi Literatur

Pada tahap ini peneliti melakukan pencarian referensi sumber data dan mempelajari materi-materi yang berhubungan dengan penelitian, seperti cara kerja algoritma LCS, pemograman python dan pengumpulan data Twitter. Referensi pembelajaran berasal dari jurnal, skripsi, dan informasi dari internet.

2. Analisis Permasalahan

Pada tahap ini peneliti melakukan analisis terhadap bahan referensi yang diperoleh pada tahap sebelumnya untuk menemukan metode dan langkah yang tepat dalam menyelesaikan permasalahan pada penelitian ini.

3. Perancangan Sistem

Pada tahap ini dilakukan perancangan arsitektur, pengumpulan data, pelatihan, dan perancangan antarmuka. Perancangan sistem dilakukan untuk menyelesaikan permasalahan yang terdapat di tahap analisis serta untuk memudahkan dalam proses implementasi. Hasil analisis dan perancangan akan diimplementasikan ke dalam sistem.

4. Implementasi

Pada tahap ini akan dilakukan proses implementasi algoritma LCS dengan menggunakan bahasa pemograman python berdasarkan perancangan program yang telah dilakukan pada tahap sebelumnya.

5. Pengujian

Pada tahap ini akan dilakukan pengujian dan analisis terhadap sistem yang sudah dibangun.

6. Dokumentasi dan Penyusunan Laporan

Pada tahap ini peneliti akan membuat dokumentasi hasil analisis dari sistem normalisasi *tweets* dan peneliti akan menyusun laporan penelitian dengan memaparkan hasil dari penelitian yang telah dilakukan.

1.7. Sistematika Penulisan

Sistematika penulisan dari penelitian ini terdiri dari lima bagian utama yaitu:

Bab 1: Pendahuluan

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

Bab 2: Landasan Teori

Bab ini berisi teori-teori yang diperlukan untuk memahami permasalahan yang dibahas pada penelitian ini. Teori-teori yang berhubungan dengan *crawling* data Twitter, *dictionary based* dan cara kerja algoritma *LCS* akan dibahas pada bab ini.

Bab 3: Analisis dan Perancangan Sistem

Bab ini berisi analisis dan penerapan *dictionary based* dan algoritma *Longest Common Subsequences* untuk melakukan proses normalisasi *tweets* yang telah dikumpulkan datanya menggunakan bahasa pemrograman python.

Bab 4: Implementasi dan Pengujian

Bab ini akan menjelaskan tentang implementasi dari perancangan penerapan yang telah dijabarkan pada bab 3. Selain itu, hasil yang didapatkan dari pengujian yang dilakukan terhadap implementasi juga dijelaskan pada bab ini.

Bab 5: Kesimpulan dan Saran

Bab ini berisi ringkasan serta kesimpulan dari rancangan yang telah dibahas serta hasil penelitian yang dijelaskan pada bab 4. Bagian akhir dari bab ini akan berisi saran yang diajukan untuk pengembangan penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1. Mikroteks

Mikroteks merupakan ruang teks yang memiliki batasan penulisan dalam cakupan kecil. Istilah mikroteks diungkapkan oleh peneliti dari US (Dela Rosa dan Ellen) pada tahun 2009 mendeskripsikan tipe penulisan dari mikroteks yang memiliki tiga karakteristik, yaitu:

1. Penulisan pada mikroteks berupa satu atau dua kalimat, dan kemungkinan sedikitnya satu kata.
2. Penulisan dengan cara yang informal dan biasanya tidak melihat struktur kalimat, dengan demikian dapat membenarkan *grammar* yang salah, pengucapan, singkatan dan akronim.
3. Termasuk *semi-structured* pada NLP.

Mikroteks sudah banyak diterapkan saat ini, terutama pada media sosial. Facebook, Twitter dan pesan SMS tergolong penulisan teks dalam cakupan kecil. Dalam menuliskan *tweets* atau pesan SMS keterbatasan karakter menjadi masalah utama sehingga pengguna dengan sengaja mempersingkat kata untuk memenuhi ruang penulisan.

2.2. Normalisasi Teks

Normalisasi teks adalah mengubah teks kalimat tidak baku menjadi teks baku 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 (Dutoit T. *Dordrecht*, 1997).

Normalisasi teks yang dibahas pada penelitian ini adalah normalisasi teks pada bagian konverter singkatan atau akronim. Bagian ini berfungsi mengubah singkatan atau akronim menjadi huruf atau deretan huruf alphabet yang menggambarkan cara pengucapannya, sebagai contoh singkatan ‘dlm’ akan diubah menjadi ‘dalam’. Adapun contoh normalisasi pada bagian lain, yaitu konverter angka yang berfungsi mengubah angka menjadi deretan huruf, contohnya angka 1220 akan diubah menjadi seribu dua ratus dua puluh dan konverter symbol atau karakter khusus, contohnya simbol ‘%’ pada kalimat ‘10%’ akan terbaca ‘sepuluh persen’. Pengubahan angka, akronim, singkatan, simbol, dan karakter khusus sangat bergantung pada basis data yang digunakan.

2.3. Dictionary Based

Dictionary based merupakan suatu model yang digunakan dalam melakukan normalisasi tanpa menggunakan algoritma. Cara kerja *dictionary based* yaitu dengan cara membaca *input* data dan membandingkannya dengan isi *dictionary*. Jika sekumpulan *string* sesuai dengan isi *dictionary* maka *output* yang dihasilkan merupakan indeks dari *dictionary* tersebut. Beberapa contoh kamus Bahasa Indonesia, yaitu:

2.3.1. KBBI (*Kamus Besar Bahasa Indonesia*)

Kamus Besar Bahasa Indonesia merupakan kamus ekabahasa resmi bahasa Indonesia. Kamus ini menjadi acuan tertinggi bahasa Indonesia baku. KBBI tersedia dalam versi *offline* dan *online*. KBBI Daring (dalam versi online atau dalam jaringan) diluncurkan pada tahun 2008 sebagai basis data yang digunakannya.

2.3.2. Kateglo

Kateglo adalah suatu layanan *web* yang menyediakan definisi, sinonim, antonim, lema bentukan, serta glosarium terkait suatu kata atau frasa. Nama Kateglo diambil dari akronim unsur layanannya, yaitu ka(mus), te(saurus), dan glo(sarium). Kateglo merupakan aplikasi sumber terbuka berbasis PHP dan berbasis data MySQL. Kateglo terdiri dari beberapa entri kamus Bahasa Indonesia dengan 72253 entri kamus, 191000 entri glosarium, 2012 entri peribahasa, serta 3423 entri singkatan dan akronim. Kateglo dapat digunakan secara manual dengan memasukan kata kunci yang dibutuhkan. Selain

itu, Kateglo juga dapat digunakan untuk aplikasi dengan mengeluarkan keluaran dalam format JSON atau XML.

2.3.3. WordNet

WordNet merupakan sebuah database kamus bahasa Inggris yang dikembangkan oleh Princeton *University*. Perbedaan antara WordNet dengan kamus bahasa pada umumnya adalah WordNet tidak hanya fokus pada kata tetapi juga memfokuskan pada makna kata. Satu makna dalam WordNet dapat dinyatakan dengan *synset* (*synonym set*). *Synonym set* merupakan kumpulan data yang mempresentasikan suatu makna. WordNet juga memberikan relasi atau hubungan antar makna seperti hipernim, holonim, meronim, dan lain-lain. Tidak hanya pada bahasa Inggris, Lab *Information Retrieval* Fakultas Ilmu Komputer Universitas Indonesia mengembangkan database WordNet yang berfokus untuk bahasa Indonesia. Pengembangan WordNet bahasa Indonesia menggunakan pendekatan *expand approach*, sehingga struktur dari WordNet bahasa Indonesia menyerupai struktur dari WordNet yang dikembangkan oleh Princeton *University*. Jumlah kata pada WordNet tidak sebanyak kamus bahasa lainnya. Saat ini, WordNet bahasa Indonesia mempunyai 1203 *synset* (*synonym set*) dan 1659 kata unik didalamnya. Jumlah relasi semantik yang dapat dibuat dari *synset* yang ada mencapai 2261 relasi.

Dari beberapa kamus tersebut, penulis memilih menggunakan kamus Kateglo sebagai acuan untuk penelitian ini, karena Kateglo lebih menyediakan apa yang dibutuhkan dalam penelitian, contohnya kamus singkatan. Pada Kateglo juga terdapat kata-kata yang sering digunakan dalam percakapan non-formal dan juga kata berimbuhan.

2.4. Twitter

Twitter merupakan media sosial dan *microblog* yang banyak digunakan untuk menyampaikan informasi, mengutarakan pendapat, ataupun mengutarakan perasaan pengguna. Twitter didirikan dan diresmikan pada tahun 2006 tepatnya pada bulan maret yang dulunya belum dibuka untuk umum, melainkan hanya untuk khusus layanan para karyawan Odeo. Twitter didirikan oleh Jack Dorsey seorang mahasiswa yang bersekolah di Universitas New York. Jejaring sosial Twitter sudah sangat dikenal oleh

setiap orang di dunia. Bahkan, di tahun 2014 Twitter menjadi salah satu dari lima besar situs yang paling sering dikunjungi oleh banyak orang.

Twitter mengalami pertumbuhan yang pesat dan dengan cepat meraih popularitas di seluruh dunia. Twitter berada pada peringkat 5 dengan pengguna sebanyak 7,2 juta. Tingginya popularitas Twitter menyebabkan layanan ini telah dimanfaatkan untuk berbagai keperluan dalam berbagai aspek, contohnya sebagai sarana protes, kampanye politik, sarana pembelajaran, dan juga sebagai media komunikasi darurat. Dengan menggunakan Twitter, pengguna bisa mengirim informasi singkat kepada orang lain yang mengikuti (*following*) akun Twitter masing-masing pengguna melalui pesan *tweet*. *Tweet* (jamak: *tweets*) adalah sebuah posting yang dibuat di situs media sosial Twitter (Oxford Dictionary, 2014). Tetapi dalam menyampaikan pesan *tweet*, media sosial ini membatasi pengguna untuk mengirim sebuah *tweet*. Pada awal didirikannya memungkinkan pengguna mengirim dan membaca pesan berbasis teks hingga 140 karakter, akan tetapi pada tanggal 07 November 2017 bertambah hingga 280 karakter.

2.5. *Natural Language Processing*

Natural Language Processing adalah salah satu bidang ilmu komputer, kecerdasan buatan, dan bahasa (linguistik) yang berkaitan dengan interaksi antara komputer dan bahasa alami manusia, seperti bahasa Indonesia atau bahasa Inggris. Tujuan utama dari studi NLP adalah membuat mesin yang mampu mengerti dan memahami makna bahasa manusia lalu memberikan respon yang sesuai.

NLP (*Natural Language Processing*) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari *user* bisa dipahami dengan baik oleh komputer. NLP memodelkan pengetahuan terhadap bahasa, baik dari segi kata, bagaimana kata-kata bergabung menjadi suatu kalimat dan konteks kata dalam kalimat. Disiplin ilmu dari NLP, yaitu:

a. Fonetik/fonologi

Fonetik atau fonologi merupakan ilmu yang berhubungan dengan suara yang menghasilkan kata yang dapat dikenali.

b. Semantik

Semantik yaitu pemetaan bentuk struktur sintaksis dengan memanfaatkan tiap kata ke dalam bentuk yang lebih mendasar dan tidak tergantung struktur kalimat. Semantik mempelajari arti suatu kata dan bagaimana dari arti kata - arti kata tersebut membentuk suatu arti dari kalimat yang utuh.

c. Pragmatik

Pengetahuan pada tingkatan ini berkaitan dengan masing – masing konteks yang berbeda tergantung pada situasi dan tujuan pembuatan sistem.

d. *Discourse knowledge*

Pengetahuan *discourse knowledge* melakukan pengenalan apakah suatu kalimat yang sudah dibaca dan dikenali sebelumnya akan mempengaruhi arti dari kalimat selanjutnya. Informasi ini penting diketahui untuk melakukan pengolahan arti terhadap kata ganti orang dan untuk mengartikan aspek sementara dari informasi.

e. *World knowledge*

Word knowledge mempelajari arti dari sebuah kata secara umum atau arti secara khusus bagi suatu kata dalam suatu percakapan dengan konteks tertentu.

Secara umum, jenis aplikasi yang dapat dibuat dalam bidang ilmu NLP terbagi dua, yaitu *text-based application* dan *dialogue-based application*. *Text-based application* adalah segala macam aplikasi yang melakukan proses terhadap teks tertulis seperti misalnya dokumen, *e-mail*, buku, dan sebagainya. Sedangkan *dialogue-based application* biasanya melibatkan bahasa lisan atau pengenalan suara, akan tetapi bisa juga memasukan interaksi dialog dengan mengetikkan teks pertanyaan melalui keyboard.

2.6. Longest Common Subsequences

2.6.1. Defenisi Longest Common Subsequences

Longest Common Subsequences adalah sebuah penyelesaian algoritma dalam mencari *common subsequence* terpanjang dari beberapa rangkaian. *Subsequence* dari sebuah *string* adalah sekumpulan karakter yang berada pada *string* yang urutan kemunculannya sama. Dalam defenisi formalnya, rangkaian Z adalah *subsequence* dari $X = \langle x_1, x_2, \dots, x_m \rangle$, jika terdapat urutan menaik $\langle i_1, i_2, \dots, i_k \rangle$ yang merupakan indeks untuk semua $j=1,2,\dots,k$, yang memenuhi $x_{i_j} = z_j$. Misalnya :

String 1 = A B C B D A B

String 2 = B D C A B A

common subsequence = B C B A

Common subsequence dari dua rangkaian adalah *subsequence* yang terdapat pada kedua rangkaian tersebut. Pada contoh diatas *common subsequence* dari dua rangkaian $S1 = ABCBDAB$ dan $S2 = BDCABA$ adalah BCBA.

2.6.2. Algoritma penyelesaian LCS

Algoritma untuk masalah LCS ini berdasarkan teorema di bawah ini :

Misal $X = \langle x_1, x_2, \dots, x_m \rangle$ dan $Y = \langle y_1, y_2, \dots, y_n \rangle$ adalah rangkaian dan $Z = \langle z_1, z_2, \dots, z_k \rangle$ adalah suatu LCS dari X dan Y .

1. Jika $x_m = y_n$ maka $z_k = x_m = y_n$ dan z_{k-1} adalah suatu LCS dari x_{m-1} dan y_{n-1}
2. Jika $x_m \neq y_n$ maka $z_k \neq x_m$ mengimplikasikan bahwa Z adalah suatu LCS dari x_{m-1} dan Y
3. Jika $x_m \neq y_n$ maka $z_k \neq y_n$ mengimplikasikan bahwa Z adalah suatu LCS dari X dan y_{n-1}

2.6.3 Pembuktian Algoritma

Adapun bukti dari teorema yang telah dijabarkan sebelumnya, yaitu:

1. Jika $z_k \neq x_m$ maka kita bisa meng-*append* $x_m = y_n$ pada Z untuk mendapatkan *common subsequence* dari X dan Y untuk panjang $k + 1$, yang kontradiksi dengan dugaan bahwa Z adalah LCS dari X dan Y. Untuk mendapatkan bahwa itu adalah LCS, yaitu misalnya $z_k = x_m = y_n$. prefix dari z_{k-1} adalah *common subsequence* dari X_{m-1} dan Y_{n-1} yang panjangnya adalah $k-1$. Lalu ada sebuah *common subsequence* W dari X_{m-1} dan Y_{n-1} dengan panjang lebih besar daripada $k-1$. Melakukan *append* $x_m = y_n$ pada W menghasilkan *common subsequence* dari X dan Y dengan panjang lebih dari k.
2. Jika $z_k \neq x_m$ maka Z adalah *common subsequence* dari x_{m-1} dan Y. Jika terdapat sebuah *common subsequence* W dari x_{m-1} dan Y yang panjangnya lebih dari k, maka hal ini berkontradiksi dengan asumsi bahwa Z adalah LCS dari X dan Y.
3. Pembuktian simetris dengan kasus 2.

2.6.4. Mendapatkan panjang LCS

Dalam mendapatkan panjang LCS harus ditentukan panjang S1 dan S2 dengan menggunakan matriks. Misal m adalah panjang S1, n adalah panjang S2, dan tab adalah matriks berukuran $m \times n$, dan $tab[i, j]$ adalah panjang LCS dari rangkaian pertama yang terdiri dari i karakter pertama S1 dengan rangkaian kedua yang terdiri dari j karakter pertama S2, maka untuk $1 \leq i \leq m$ dan $1 \leq j \leq n$, sesuai dengan fungsi rekursif di atas, maka :

$$tab[i, j] = \begin{cases} tab[i - 1, j - 1] + 1, & S1[i] = S2[j] \\ \max (tab[i - 1, j], tab[i, j - 1]) & \end{cases}$$

Misal, untuk S1 = FTKP dan S2 = FOTOKOPI tabel yang terbentuk adalah tabel berukuran 5×9 yang isinya sebagai berikut :

Tabel 2.1. Hasil kalkulasi LCS dari FTKP dan FOTOKOPI

		F	O	T	O	K	O	P	I
		1	2	3	4	5	6	7	8
F	1	1	1	1	1	1	1	1	1
T	2	1	1	2	2	2	2	2	2
K	3	1	1	2	2	3	3	3	3
P	4	1	1	2	2	3	3	4	4

Sesuai dengan arti notasi dari setiap $tab[i,j]$ bahwa LCS dari S1 dari S2 adalah $tab[m,n]$ adalah 4.

2.6.5. Mendapatkan LCS

Dalam mendapatkan LCS, kita bisa merunut balik tabel dimulai dari $tab[m, n]$. Dalam runutbalik ini, yang kita lakukan sebenarnya hanyalah merunutbalik pilihan yang dilakukan pada saat kalkulasi $tab[i, j]$.

Untuk setiap $1 \leq i \leq m$ dan $1 \leq j \leq n$, jika $S1[i]$ sama dengan $S2[j]$, maka karakter itu pasti terdapat pada LCS. Selain itu, cek darimana mendapatkan LCS saat itu, lakukan pemilihan yang sama. Lakukan hal tersebut dimulai dari $i = m$ dan $j = n$.

Setelah hal itu dilakukan maka akan terbentuk tabel di bawah ini. Di mana *cell* yang dihitamkan menandakan karakter yang masuk ke dalam LCS.

Tabel 2.2. Hasil untuk mendapatkan LCS dari FTKP dan FOTOKOPI

		F	O	T	O	K	O	P	I
		1	2	3	4	5	6	7	8
F	1	1	1	1	1	1	1	1	1
T	2	1	1	2	2	2	2	2	2
K	3	1	1	2	2	3	3	3	3
P	4	1	1	2	2	3	3	4	4

Berdasarkan tabel tersebut di atas terlihat bahwa LCS dari FTKP dan FOTOKOPI adalah FTKP.

2.7. Penelitian Terdahulu

Penelitian normalisasi mikroteks sudah pernah dilakukan sebelumnya oleh peneliti luar yaitu Xue Zhenzhen, *et al.* (2011). Penelitian tersebut melakukan normalisasi mikroteks dengan menggunakan data SMS dan Twitter. Pada penelitian tersebut dijelaskan ada empat jenis faktor dalam melakukan normalisasi mikroteks yaitu ortografi, fonetik, kontekstual, dan akronim. Dengan empat faktor tersebut mereka melakukan normalisasi dengan menggunakan pendekatan *Channel Models* pada masing-masing faktor. *Grapheme Channel* pada ortografi, *Phoneme Channel* pada fonetik, *Context Channel* pada kontekstual, dan *Acronym Channel* pada akronim. Penelitian ini menghasilkan bahwa pendekatan *Channel Model* lebih baik digunakan pada data SMS *public* dibandingkan Twitter.

Pada tahun 2015 penelitian dilakukan oleh Khoury Richard (2015) yaitu melakukan normalisasi mikroteks pada sosial media menggunakan algoritma Levenshtein distance. Penelitian ini melakukan normalisasi pada masalah fonetik berdasarkan struktur OOV dalam Bahasa Inggris dan juga melakukan normalisasi untuk singkatan. Hasil dari normalisasi mendapatkan akurasi sekitar 60% dari keseluruhan masalah sedangkan akurasi untuk jenis normalisasi tertentu menghasilkan jangkauan hingga 80% sampai 90%.

Penelitian selanjutnya dilakukan oleh Irawan Johaness (2016) yaitu melakukan normalisasi pada teks Twitter dengan menggunakan *Noisy Channel Model*. *Noisy Channel Model* menjelaskan bahwa bentuk yang terlihat pada suatu kata yang terlihat dapatlah menjadi sebuah bentuk yang terdistorsi dari bentuk aslinya. Hasil dari penelitian ini, sistem mampu membedakan beberapa variasi dari sebuah kata, namun tidak dapat membedakan kata yang harusnya tidak dinormalisasi dan tidak dapat memperbaiki singkatan. Akurasi yang didapatkan sistem adalah 70,12% dan F1 Score sebesar 38,36%.

Pada tahun 2017 penelitian dilakukan oleh Saragih Tri (2017) yaitu melakukan normalisasi teks Twitter dengan menggunakan algoritma jarak *string* pada pemrograman R. Penelitian ini membandingkan 10 metode *stringdist* untuk menghitung jumlah

perbedaan jarak antar *string* yang selanjutnya digunakan untuk pengubahan *string*. Metode yang digunakan yaitu *lv* (*Levenshtein distance*), *osa* (*Optimal string alignment*), *dl* (*Full damerau levenshtein distance*), *hamming distance*, *lcs* (*Longest common substring distance*), *qgram* (*q-gram distance*), *cosine* (*Cosine distance*), *jaccard* (*Jaccard distance*), *jw* (*Jaro or Jaro-Winker distance*), dan *soundex* (*distance based on soundex encoding*). Berdasarkan hasil perbandingan 10 metode, akurasi yang paling tinggi adalah menggunakan metode *Longest Common Substring distance* yaitu 69%. Perubahan yang dilakukan tidak semuanya benar sesuai dengan yang seharusnya, seperti masih terdapat kesalahan kata yang ditampilkan jika kata tersebut mengandung imbuhan karena tidak semua kata yang mengandung imbuhan ada pada kamus. Selain itu disebabkan oleh adanya saran kata lain yang memiliki jarak minimal yang sama dan tidak adanya kata dengan perbaikan seharusnya pada kamus.

Penelitian selanjutnya dilakukan oleh Satapathy Ranjan, *et al.* (2017) melakukan normalisasi mikroteks untuk kata dasar Bahasa Inggris dan meningkatkan akurasi klasifikasi dari analisis sentimen pada Twitter. Penelitian ini menggunakan metode kombinasi untuk normalisasi mikroteks, yaitu *Lexicon Based* dan *Phonetic Based*. Hasil dari penelitian ini menunjukkan adanya indeks kesamaan (> 0.8) antara *tweet* yang telah dinormalisasi oleh sistem dengan *tweet* yang dinormalisasi secara manual sebanyak 85.31% dan terdapat peningkatan akurasi sebanyak $>4\%$ dalam pendeteksian polaritas setelah normalisasi.

Selanjutnya, pada tahun 2017 penelitian juga dilakukan oleh Hanafiah Novita, *et al.* (2017). Penelitian ini melakukan normalisasi teks pada Twitter untuk *complaint category*. Penelitian ini memanfaatkan data Twitter khususnya Bahasa Indonesia. Proses dibagi dalam tiga tahap, yaitu proses pembersihan, deteksi OOV, dan penggantian kata. Penggunaan corpus dan penerapan algoritma Levensthein dimanfaatkan dalam peneltian ini untuk menangani masalah singkatan. Penerapan algoritma Levensthein berhasil mencapai tingkat akurasi sekitar 90% dalam *complaint category*.

Rangkuman dari penelitian terdahulu yang dijelaskan di atas dapat dilihat pada Tabel 2.3. berikut :

Tabel 2.3. Penelitian Terdahulu

No.	Judul	Peneliti (tahun)	Metode	Keterangan
1	<i>Normalizing Microtext</i>	Xue Zhenzhen, <i>et al.</i> (2011)	<i>Multi-Channel Model</i>	Penelitian ini melakukan normalisasi mikroteks pada SMS dan Twitter dengan melihat empat faktor penulisan yaitu ortografi, fonetik, kontekstual, dan singkatan.
2	<i>Microtext Normalization using Probably-Phonetically-Similar Word Discovery</i>	Khoury Richard (2015)	<i>Levenshtein distance</i>	Penelitian ini melakukan normalisasi mikroteks pada masalah fonetik dan singkatan dalam Bahasa Inggris dengan hasil akurasi keseluruhan 60%.
3	Normalisasi Teks Twitter Bahasa Indonesia Berbasiskan <i>Noisy Channel Model</i>	Irawan Yohanes (2016)	<i>Noisy Channel Model</i>	Pada penelitian ini, sistem mampu membedakan beberapa variasi dari sebuah kata. Akurasi yang didapatkan sistem adalah 70,12% dan F1 <i>Score</i> sebesar 38,36%.

Tabel 2.3 Penelitian Terdahulu (Lanjutan)

4	Normalisasi Teks Pada Teks Twitter Berbahasa Indonesia Menggunakan Algoritme Jarak <i>String</i> Pada <i>R</i>	Saragih Tri (2017)	Algoritma Jarak <i>String</i> Pada <i>R</i>	Penelitian ini menggunakan perbandingan 10 metode <i>stringdist</i> . Dari perbandingan tersebut akurasi tertinggi dihasilkan oleh <i>lcs</i> dengan nilai 69%.
5	<i>Phonetic-Based</i> <i>Microtext</i> <i>Normalization</i> <i>for Twitter</i> <i>Sentiment</i> <i>Analysis</i>	Satopathy Ranjan, <i>et al.</i> (2017)	<i>Lexical based</i> <i>dan Phonetic</i> <i>Based</i>	Penelitian ini melakukan normalisasi mikroteks untuk kata dasar Bahasa Inggris dan meningkatkan akurasi klasifikasi dari analisis sentimen pada Twitter.
6	<i>Text</i> <i>Normalization</i> <i>Algorithm on</i> <i>Twitter in</i> <i>Complaint</i> <i>Category</i>	Hanafiah Novita, <i>et al.</i> (2017)	<i>Levensthein</i> <i>distance</i>	Penelitian ini melakukan normalisasi teks Twitter untuk <i>complaint category</i> menggunakan algoritma Levensthein dengan tingkat akurasi 90%.

Perbedaan penelitian yang dilakukan penulis dengan penelitian terdahulu adalah pada penelitian ini penulis menggunakan algoritma yang berbeda yaitu algoritma *Longest Common Subsequences*. Dari penelitian terdahulu yang didapat belum ada yang menggunakan algoritma tersebut.

BAB 3

ANALISIS DAN PERANCANGAN

3.1. Data yang Digunakan

Data yang digunakan untuk penelitian ini menggunakan data *tweets* dari Twitter. Pengumpulan *tweets* dilakukan dengan cara *crawling* data *tweets* menggunakan API Twitter. Data *tweets* dikumpulkan berdasarkan kata kunci (*keyword*). Pemilihan kata kunci dipilih sesuai dengan kebutuhan dan tidak memiliki kriteria tertentu. Contoh beberapa *keyword* yang digunakan dalam penelitian ini yaitu ‘berita’, ‘yg’, ‘politik’, ‘medan’, dan lain sebagainya. Data yang dihasilkan berjumlah 400 *tweets* dengan berbagai *keyword* dalam format *json*. Contoh beberapa *tweets* hasil *crawling* dalam bentuk *json* yang mengandung singkatan dalam penulisannya yaitu:

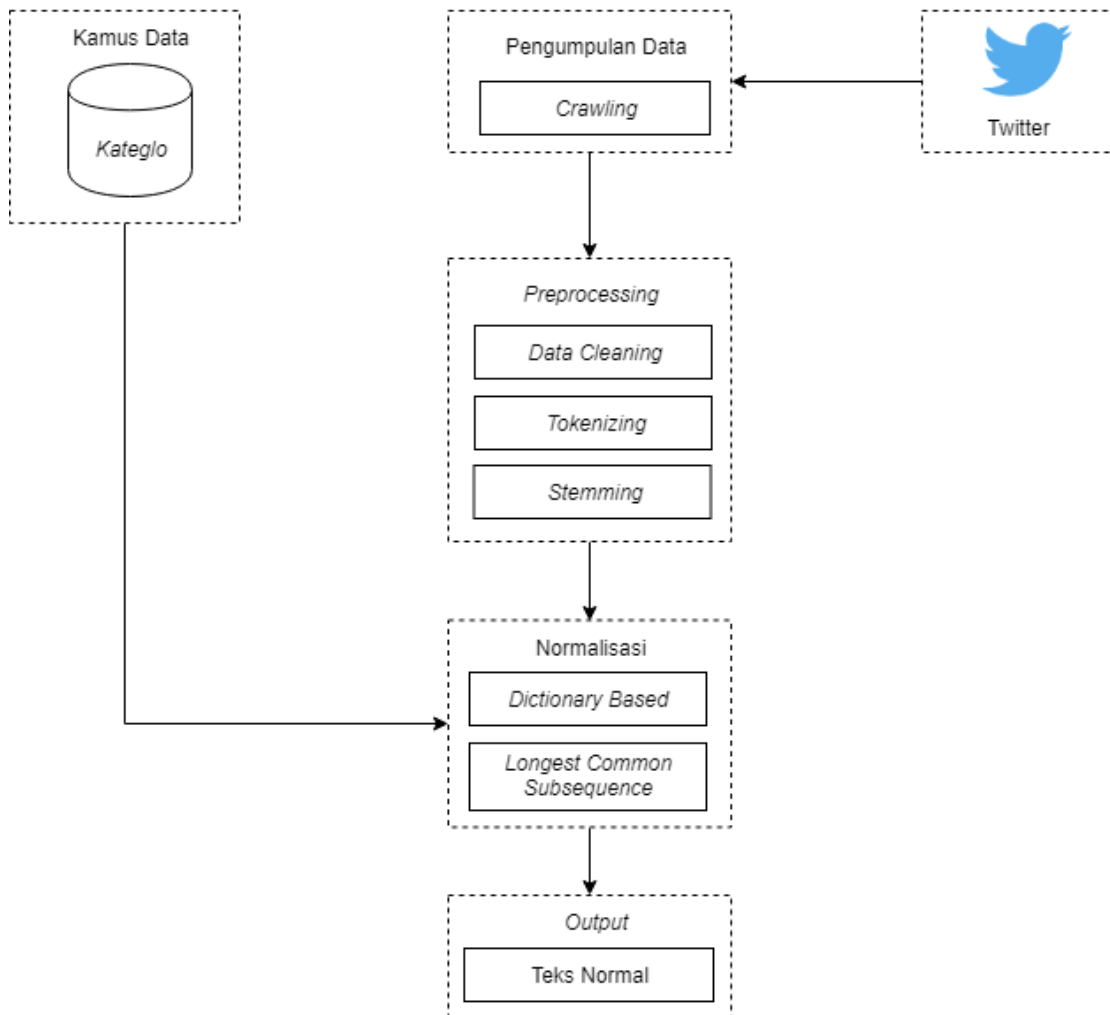
[[“akhirnya dia menikah juga, slmt pengantin bru..”], ["awas jgn tertipu jebakan oposisi dgn hembuskan ingin jemput buron"], [“slma ini aku hnya bisa menangis!”], ["duit tak mngizinkan 3"], ["kalo lg gini kgn shbt gua nih yang belabelain dengerin gua nangis ditelfon bawain makanan kkosan ngobrol gajelas"]]

3.2. Perancangan Sistem

3.2.1. Arsitektur umum

Metode yang diajukan untuk penelitian ini terdiri dari beberapa langkah. Setelah data terkumpul, akan dilakukan tahap *pre-processing* terlebih dahulu yaitu seperti *data cleaning* (pembersihan data), *tokenizing* (pemecahan karakter), dan proses *stemming*. Setelah langkah-langkah *pre-processing* selesai, maka data akan siap untuk diproses. Proses tersebut yaitu proses normalisasi teks dengan menggunakan *dictionary based* dan algoritma *Longest Common Subsequences*. Setiap tahapan tersebut akan dijelaskan

secara rinci pada bagian selanjutnya. Adapun arsitektur umum yang menggambarkan setiap tahapan pada penelitian ini dapat dilihat pada Gambar 3.1:



Gambar 3.1. Arsitektur Umum

Penjelasan dari tahapan-tahapan yang ada pada Gambar 3.1 akan diuraikan secara detail sebagai berikut:

1. Pengumpulan Data

Pengumpulan data merupakan proses yang digunakan dalam mengumpulkan informasi untuk menganalisis beberapa masalah. Data pada penelitian ini adalah data kamus dan data *tweets* yang diambil melalui proses *crawling* dalam bentuk *json*.

Data kamus diperoleh dari hasil *crawling* database Kateglo. Contoh data kamus dapat dilihat pada Lampiran 1 dan Lampiran 2. Pengambilan data *tweets* juga diperoleh dari hasil *crawling* dengan menggunakan *API* Twitter. Sebelumnya harus melakukan pendaftaran atau masuk ke akun pribadi pada Twitter *apps*. Setelah masuk, akan ditampilkan *API key*, *API secret*, *access tokens*, dan *access tokens secret* untuk menghubungkan ke pemrograman python. Tetapi, hasil dari *crawling* tersebut harus melalui *pre-processing* terlebih dahulu untuk dapat diproses selanjutnya, karena masih terdapat beberapa *string* yang harus dibuang seperti tanda baca, *hashtag*, dan hal lain yang tidak diperlukan untuk proses normalisasi.

2. *Pre-processing*

Dalam *pre-processing* terdiri dari beberapa tahap, yaitu :

a. *Data Cleaning*

Data cleaning merupakan pembersihan data atau penghapusan data (*data scrubbing*) untuk memastikan sekumpulan data sudah benar dan akurat. Pada kasus ini akan dilakukan pembersihan dan penghapusan karakter seperti penghapusan tanda baca, *link*, *hashtag*, dan lainnya yang tidak diperlukan untuk proses normalisasi. Contoh data *tweet* yang mengandung *hashtag* dan *retweet*, yaitu:

Tweets sebelum dilakukan pembersihan:

```
[["rt akhirnya dia menikah juga slmt pengantian baru"], ["awas jgn tertipu jebakan oposisi dgn hembuskan ingin jemput #buron"], ["duit tak mngizinkan #3"], ["rt kalo lg gini kgn shbt gua nih yang belabelain dengerin gua nangis ditelfon bawain makanan kkosan ngobrol gajelas"]]
```

Tweets setelah dilakukan pembersihan:

```
[["akhirnya dia menikah juga slmt pengantian baru"], ["awas jgn tertipu jebakan oposisi dgn hembuskan ingin jemput buron"], ["duit tak mngizinkan 3"], ["kalo lg gini kgn shbt gua nih yang belabelain dengerin gua nangis ditelfon bawain makanan kkosan ngobrol gajelas"]]
```

b. *Tokenizing*

Tokenizing merupakan langkah untuk memotong dokumen menjadi potongan-potongan kecil yang disebut token dan terkadang disertai langkah untuk membuang karakter tertentu seperti tanda baca (Manning, Raghavan, dan Schutze, 2009). Selain itu spasi digunakan untuk memisahkan antar kata tersebut. Dalam sebuah *tweet* tidak hanya terdapat satu kalimat, namun terdapat dua atau lebih kalimat. Pada penelitian ini proses *tokenizing* diperlukan untuk memisahkan kalimat per-kalimat dalam satu *tweet* agar mempermudah proses normalisasi. Contoh *tweets* yang sudah melalui proses *tokenizing* dengan memisahkan kata per-kata dalam satu *tweet*, yaitu:

Tweets sebelum mengalami proses *tokenizing*:

[[“akhirnya dia menikah juga slmt pengantian baru”], [“awas jgn tertipu jebakan oposisi dgn hembuskan ingin jemput buron”], [“duit tak mngizinkan 3”], [“slma ini aku hnya bisa menangis”], [“kalo lg gini kgn shbt gua nih yang belabelain dengerin gua nangis ditelfon bawain makanan kkosan ngobrol gajelas”]]

Tweets setelah mengalami proses *tokenizing*:

[[“akhirnya”, ”dia”, “menikah”, “juga”, “slmt”, “pengantin”, “baru”], [“awas”, “jgn”, “tertipu”, “jebakan”, “oposisi”, “dgn”, “hembuskan”, “ingin”, “jemput”, “buron”], [“slma”, “ini”, “aku”, “hnya”, “bisa”, “menangis”], [“duit”, “tak”, “mngizinkan”, “3”], [“kalo”, “lg”, “gini”, “kgn”, “shbt”, “gua”, “nih”, “yang”, “belabelain”, “dengerin”, “gua”, “nangis”, “ditelfon”, “bawain”, “makanan”, “kkosan”, “ngobrol”, “gajelas”]]

c. *Stemming*

Stemming bertujuan untuk mentransformasikan sebuah kata menjadi kata dasar (*root*) dengan menghilangkan semua imbuhan kata yang terdapat pada awalan, sisipan, ataupun akhiran. Pada penelitian ini proses *stemming* dibutuhkan untuk mengecek kata dasar pada kamus data dari kata masukan yang mengandung imbuhan. Jika kata dasar masukan terdapat pada kamus data artinya kata masukan tersebut sudah benar, maka kata masukan tersebut tidak perlu diproses lagi menggunakan *dictionary based* ataupun algoritma. Proses *stemming* bertujuan untuk mempertahankan kebenaran kata masukan yang mengandung

imbuhan tetapi kata masukan tersebut tidak terdapat dalam kamus. Langkah-langkah yang dilakukan dalam proses *stemming* adalah sebagai berikut.

- Cek kata masukan dalam kamus, jika kata terdapat dalam kamus, maka kata tersebut sudah merupakan kata dasar dan jika tidak ada maka kata tersebut akan dicek ketahap selanjutnya.
- Cek awalan dan akhiran pada kata masukan, jika terdapat awalan ataupun akhiran, maka kata masukan tersebut merupakan kata berimbuhan. Pada proses ini akan memisahkan awalan dan akhiran untuk mendapatkan kata dasar. Kemudian cek kata dasar dalam kamus, jika ada maka kata yang mengandung imbuhan tersebut merupakan kata masukan yang sudah benar. Jika tidak ada, maka proses akan berlanjut ke tahap selanjutnya.
- Jika kata masukan merupakan singkatan, maka proses *stemming* akan berhenti dan akan berlanjut ke proses normalisasi.

Contoh *tweet* yang mengandung imbuhan dan harus melewati proses *stemming*, yaitu :

['sudah' 'jgn' 'memulai' 'lagi' 'yg' 'sudah' 'diakhiri' 'jalani' 'saja' 'hidup' 'baru']

Pada *tweet* tersebut terdapat beberapa kata yang mengandung imbuhan yaitu 'memulai', 'diakhiri', dan 'jalani'. Beberapa kata tersebut harus terlebih dahulu dicek kebenaran katanya menggunakan proses *stemming* agar mendapatkan hasil yang maksimal, seperti ditunjukkan pada Tabel 3.1.

Tabel 3.1. Pengecekan *stemming*

kata	awalan	Akhiran	kata dasar
memulai	me	i	mulai
diakhiri	di	i	akhir
jalani	-	i	jalan

Kalimat setelah pengecekan *stemming*:

['sudah' 'jgn' '(me)mulai' 'lagi' 'yg' 'sudah' '(di)akhir(i)' 'jalan(i)' 'saja' 'hidup' 'baru']

Pada pengecekan *stemming*, jika kata masukan yang mengandung imbuhan merupakan kata dasar sesuai dengan kamus data, maka artinya kata masukan tersebut merupakan kata baku yang sudah benar. Oleh karena itu, kata masukan tersebut tidak akan mengalami perubahan dan tidak perlu melewati proses normalisasi. Maka, *tweet* tersebut akan dikembalikan ke bentuk semula untuk diproses selanjutnya.

3. Normalisasi

Pada proses normalisasi ada dua metode yang dilakukan, yaitu dengan menggunakan *dictionary based* dan algoritma *Longest Common Subsequences* dengan acuan kamus data. Kamus data yang digunakan sebagai acuan yaitu kamus data Kateglo yang terdiri atas beberapa entri. Pada penelitian ini hanya akan menggunakan entri kamus secara keseluruhan dan kamus khusus singkatan/akronim.

a. *Dictionary based*

Pada kamus data Kateglo tidak hanya terdapat entri kamus, tetapi ada beberapa entri lainnya seperti entri glosarium, entri pribahasa, serta entri singkatan dan akronim. Proses *dictionary based* akan memanfaatkan kamus khusus singkatan. Kamus entri singkatan pada Kateglo berisi sebanyak 7000 singkatan dan akronim. Singkatan tersebut merupakan singkatan kata baku sesuai Kamus Besar Bahasa Indonesia dan juga singkatan kata yang umum sering digunakan sehari-hari oleh masyarakat. Contoh isi kamus tersebut dapat dilihat pada Lampiran 2.

Pada proses ini jika masukan sudah terdapat pada entri singkatan atau akronim di Kateglo, maka proses dari masukan tersebut akan memberikan saran kata sesuai dengan entri yang terdapat pada kamus. Misal, terdapat singkatan kata dari sebuah *tweet* yaitu “tdk”, maka sistem akan terlebih dahulu mengecek kedekatan *string* pada kamus singkatan. Setelah dicek pada kamus singkatan, jika masukan tersebut ada dalam entri kamus singkatan, yaitu “tidak” maka proses akan berhenti. Tetapi, jika masukan tidak terdapat dalam kamus singkatan maka proses normalisasi akan dilanjutkan menggunakan perhitungan algoritma. Dengan menerapkan proses *dictionary based* akan memaksimalkan hasil normalisasi *tweet*. Contoh *tweet* yang mengandung singkatan dan beberapa dari kata masukan tersebut terdapat dalam kamus singkatan, yaitu:

['rezeki', 'yg', 'sebenar', 'itu', 'ialah', 'yg', 'tdk', 'nmpk', 'pd', 'mata', 'tp', 'terasa', 'pd', 'hati']

Pada *tweet* tersebut terdapat kata masukan yang mengandung singkatan, yaitu 'yg', 'tdk', 'nmpk', 'pd', dan 'tp'. Proses *dictionary based* akan mengecek kata masukan tersebut dalam kamus singkatan. Setelah dicek pada kamus singkatan, beberapa kata masukan terdapat dalam kamus, seperti terlihat pada Tabel 3.2.

Tabel 3.2. *Dictionary based*

Kata masukan	Dictionary
yg	yang
tdk	tidak
nmpk	-
pd	pada
tp	tapi

Hasil *tweet* dari normalisasi menggunakan *dictionary based*, yaitu :

['rezeki', 'yang', 'sebenar', 'itu', 'ialah', 'yang', 'tidak', 'nmpk', 'pada', 'mata', 'tapi', 'terasa', 'pada', 'hati']

b. Longest Common Subsequences

Pada entri singkatan di Kateglo tidak semua masukan yang termasuk singkatan kata terdapat pada entri. Oleh karena itu pada tahap ini akan dilakukan normalisasi singkatan kata menggunakan algoritma *Longest Common Subsequences*.

Pada penelitian ini untuk mendapatkan panjang LCS dari dua buah *string* digunakan perhitungan matriks $m \times n$ dan $tab[i,j]$, dengan m adalah panjang *string* pertama dan n adalah panjang *string* kedua. Misal terdapat masukan *tweet* “akhirnya dia menikah juga slmt pengantin baru”. Pada *tweet* tersebut terdapat sebuah kata baku yang disingkat yaitu kata “slmt”. Algoritma LCS akan mencari kedekatan kata yang memiliki *string* yang sama dengan masukan yaitu “slmt” dengan menggunakan perhitungan matriks tersebut. Untuk tidak membuat sistem mengulang kerja diawal, maka dalam proses pencarian *string* akan langsung dimulai sesuai huruf awalan pada

kata masukan tersebut. Jika masukan “slmt”, huruf awalnya adalah ‘s’, maka proses pencarian akan langsung dilakukan pada kata yang berawalan huruf ‘s’. Perhitungan matriks akan mencari kedekatan kata dengan membandingkan huruf per-huruf yang ada pada kata masukan dengan kata yang terdapat pada kamus. Algoritma LCS akan mencari kata pada kamus yang memiliki huruf sesuai dengan kata masukan, yaitu ‘s’, ‘l’, ‘m’, dan ‘t’. Pada kamus Kateglo terdapat beberapa kedekatan kata dari “slmt”, yaitu “selamat”, “selimut”, “selomot”, “selampit”, “selimpat” dan sebagainya. Untuk memastikan kedekatan kata tersebut maka akan dibuat tabel perhitungan matriks sebagai berikut.

a. selamat

		s	e	l	a	m	a	t
		1	2	3	4	5	6	7
s	1	1	1	1	1	1	1	1
l	2	1	1	2	2	2	2	2
m	3	1	1	2	2	3	3	3
t	4	1	1	2	2	3	3	4

Berdasarkan perhitungan tabel matriks diatas, maka dapat diketahui bahwa LCS dari *string* masukan dan *string* keluaran memiliki kedekatan string yang sama, seperti terlihat perbandingan dibawah ini:

S1 = S L M T

S2 = S E L A M A T

common subsequence = S L M T

b. selimut

		s	e	l	i	m	u	t
		1	2	3	4	5	6	7
s	1	1	1	1	1	1	1	1
l	2	1	1	2	2	2	2	2
m	3	1	1	2	2	3	3	3
t	4	1	1	2	2	3	3	4

Perbandingan S1 dan S2:

S1 = S L M T

S2 = S E L I M U T

common subsequence = S L M T

c. selomot

		s	e	l	o	m	o	t
		1	2	3	4	5	6	7
s	1	1	1	1	1	1	1	1
l	2	1	1	2	2	2	2	2
m	3	1	1	2	2	3	3	3
t	4	1	1	2	2	3	3	4

Perbandingan S1 dan S2:

S1 = S L M T

S2 = S E L O M O T

common subsequence = S L M T

d. selampit

		s	e	l	a	m	p	i	t
		1	2	3	4	5	6	7	8
s	1	1	1	1	1	1	1	1	1
l	2	1	1	2	2	2	2	2	2
m	3	1	1	2	2	3	3	3	3
t	4	1	1	2	2	3	3	3	4

Perbandingan S1 dan S2:

S1 = S L M T

S2 = S E L A M P I T

common subsequence = S L M T

e. selimpat

		s	e	l	i	m	p	a	t
		1	2	3	4	5	6	7	8
s	1	1	1	1	1	1	1	1	1
l	2	1	1	2	2	2	2	2	2
m	3	1	1	2	2	3	3	3	3
t	4	1	1	2	2	3	3	3	4

Perbandingan S1 dan S2:

S1 = S L M T

S2 = S E L I M P A T

common subsequence = S L M T

Keterangan : *cell* yang dihitamkan untuk urutan pertama menandakan karakter yang masuk dalam LCS, yaitu 's', 'l', 'm', dan 't'.

Banyak kemungkinan kata lain pada kamus yang memiliki *string* sama dan jumlah huruf yang sama dengan kata masukan. Tetapi, dari pengamatan secara manual, penulis membuat suatu asumsi bahwa LCS dengan jumlah karakter yang terpendek merupakan kata yang dianggap benar sesuai konteks dari kalimat. Penulis mengamati sekitar 100 kata yang dihasilkan dari LCS untuk mendukung asumsi tersebut. Keseluruhan kata tersebut dapat dilihat pada halaman Lampiran 4.

Setiap kata akan dilakukan pengecekan jumlah karakter. Misalnya terdapat suatu kalimat “kalau ikhlas mengapa hrus teriak”. Pada kalimat tersebut terdapat beberapa kata yang mengandung singkatan, yaitu “hrus”. Pada LCS kata masukan “hrus” menghasilkan beberapa kata yaitu “harus” dengan jumlah karakter 5, “humerus” dengan jumlah karakter 7, dan “hidraulis” dengan jumlah karakter 9. Adapun kata dengan jumlah karakter terpendek yaitu “harus” dan kata tersebut juga merupakan hasil kata yang tepat sesuai konteks kalimat.

Pada Tabel 3.3 berikut akan menampilkan beberapa kata dari 100 kata uji yang dihasilkan dari LCS.

Tabel 3.3. Contoh Beberapa Kata Uji Hasil LCS

Singkatan	Hasil LCS	Jumlah Karakter	Keterangan
mngapa	mengapa	7	Benar
	mengapai	8	Salah
	mengapam	8	Salah
hrus	harus	5	Benar
	humerus	7	Salah
	hidraulis	9	Salah
teriak	teriak	6	Benar
	teriakan	8	Salah
	terinjak	8	Salah
tlah	talh	5	Salah
	telah	5	Benar
	tulah	5	Salah
	telaah	6	Salah

sblm	sublim	6	Salah
	sebelum	7	Benar
	sublema	7	Salah
sy	saya	4	Benar
	sayu	4	Salah
	syah	4	Salah
bgitu	begitu	6	Benar
	begituan	8	Salah
	begitu pun	9	Salah
byran	bayaran	7	Benar
	berkebyaran	11	Salah
	berkeluyuran	12	Salah
mkn	makan	5	Benar
	makin	5	Salah
	makna	5	Salah
utk	utik	4	Salah
	untuk	6	Benar
	utrikel	7	Benar

Dari 100 kata yang diamati, 90% mendukung pernyataan bahwa LCS dengan kata terpendek merupakan kata yang tepat. Oleh karena itu, berdasarkan hal tersebut diasumsikan bahwa kata dengan jumlah karakter terpendek adalah kata yang tepat sesuai konteks dari kalimat.

c. Output

Output yang dihasilkan dari proses normalisasi menggunakan *dictionary based* dan algoritma LCS adalah *tweet* dengan hasil singkatan kata yang sebenarnya yang terdapat dalam kamus. Tetapi, ada beberapa hasil dari kata masukan yang tidak sesuai dengan struktur kalimat, karena dalam penelitian ini tidak membahas atau melihat kesenjangan struktur kalimat. *Output* yang akan ditampilkan pada sistem adalah hasil singkatan kata dari saran kata sesuai kamus singkatan atau hasil dari perhitungan algoritma dengan jumlah karakter terpendek. Tetapi, jika pada hasil normalisasi terdapat pilihan kata yang memiliki jumlah karakter yang sama, maka

output dari sistem ini hanya akan menampilkan saran kata atau pilihan kata dengan jumlah karakter yang sama sesuai urutan abjad. Contohnya yaitu:

Tweet sebelum dinormalisasi:

['slmt', 'yg', 'sudah', 'menempuh', 'hdp', 'bru', 'smoga', 'bahagia']

Pada penelitian ini, sistem akan menampilkan *word choices* dari kata masukan yang mengandung singkatan. *Word choices* berfungsi untuk melihat perbandingan dari saran kata yang diberikan dan untuk melihat jumlah karakter setiap kata, seperti terlihat pada Tabel 3.4 berikut:

Tabel 3.4. *Word Choices*

	<i>Word Choices</i>					
	<i>Algorithm</i>					<i>Dict</i>
slmt	selamat	selimut	selomot	selampit	selimpat	-
	7	7	7	8	8	-
yg	yang	yoga	yogi	yogia	yargon	yang
	4	4	4	5	6	-
hdp	hadap	hidup	hadapan	hidropati	hidroskop	-
	5	5	7	9	9	-
bru	baru	beru	biru	bruk	buru	-
	4	4	4	4	4	-
smoga	semoga	seismograf	seismogram	sfigmograf	sinematograf	-
	6	10	10	10	12	-

Pada Tabel *word choices* tersebut, terlihat beberapa saran kata dengan jumlah karakter terpendek atau dengan jumlah karakter yang sama. Hasil yang akan

ditampilkan adalah *word choices* dengan jumlah karakter terpendek. Jika terdapat jumlah karakter terpendek lebih dari satu saran kata, maka semua saran kata tersebut merupakan *output* dari kata masukan.

Contoh *tweet* sebelum dinormalisasi:

‘slmt yg sudah menempuh hdp bru smoga bahagia’

Setelah mendapatkan beberapa pilihan kata, maka sistem akan menampilkan pilihan kata dengan jumlah karakter terpendek atau jumlah karakter terpendek yang sama.

Contoh *tweet* setelah dinormalisasi:

‘(selamat | selimut | selomot) (yang) sudah menempuh (hadap | hidup)
(baru | beru | biru | bruk | buru) (semoga) bahagia’

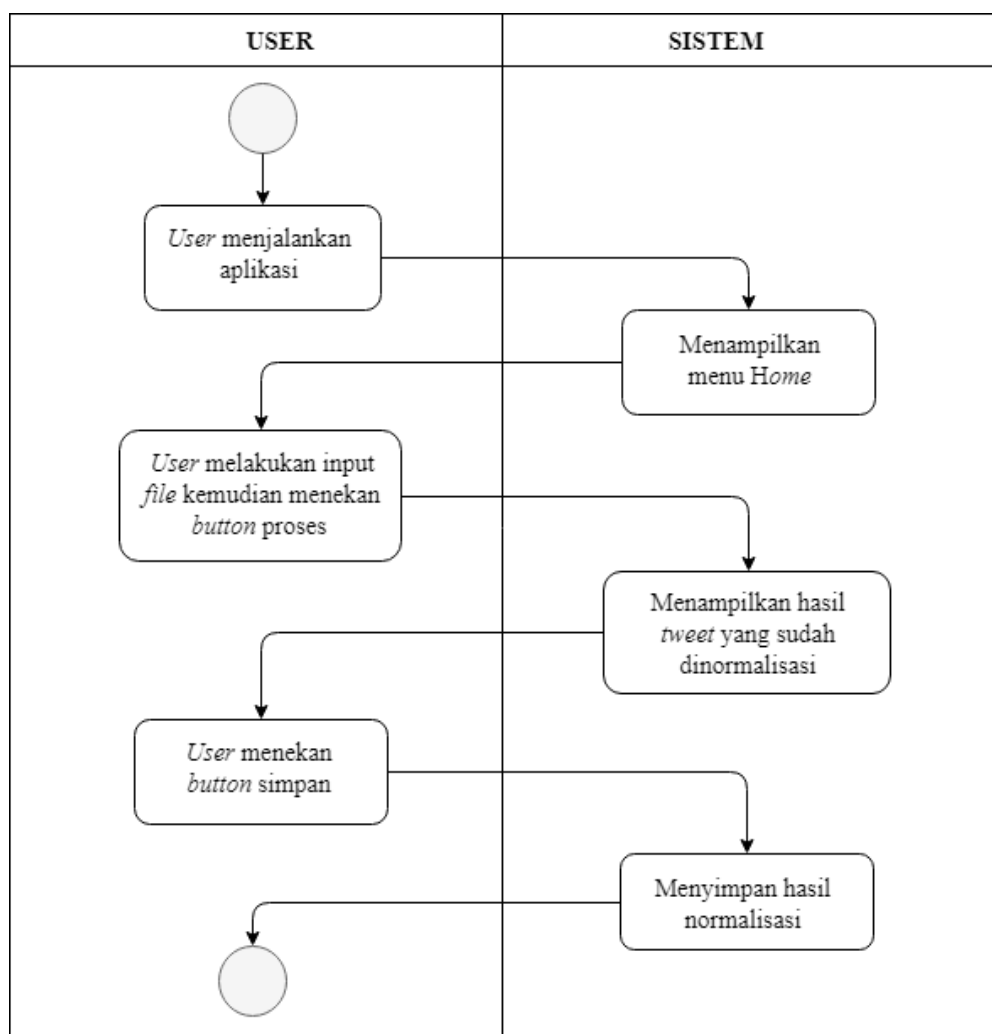
Pada penelitian ini, sistem hanya dapat memberikan saran kata jika terdapat lebih dari satu pilihan saran kata dengan jumlah karakter terpendek yang sama. Penelitian ini tidak dapat menentukan satu dari beberapa pilihan saran kata, karena penelitian ini tidak melihat hubungan atau struktur kalimat.

Setelah melakukan perbaikan singkatan kata ke bentuk kata normal, maka diperlukan evaluasi normalisasi teks untuk mengevaluasi fungsi terhadap *output* yang dihasilkan. Caranya adalah dengan membandingkan hasil normalisasi menggunakan algoritma LCS dengan hasil normalisasi manual.

3.2.2. Diagram aktivitas sistem

Diagram aktivitas dijelaskan oleh Yuni Sugiarti (2013) bahwa diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis yang sedang dirancang dan bagaimana masing-masing aliran berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir. Diagram aktivitas pada penelitian ini yaitu diagram aktivitas normalisasi yang akan menggambarkan aktivitas yang dikerjakan pada menu normalisasi.

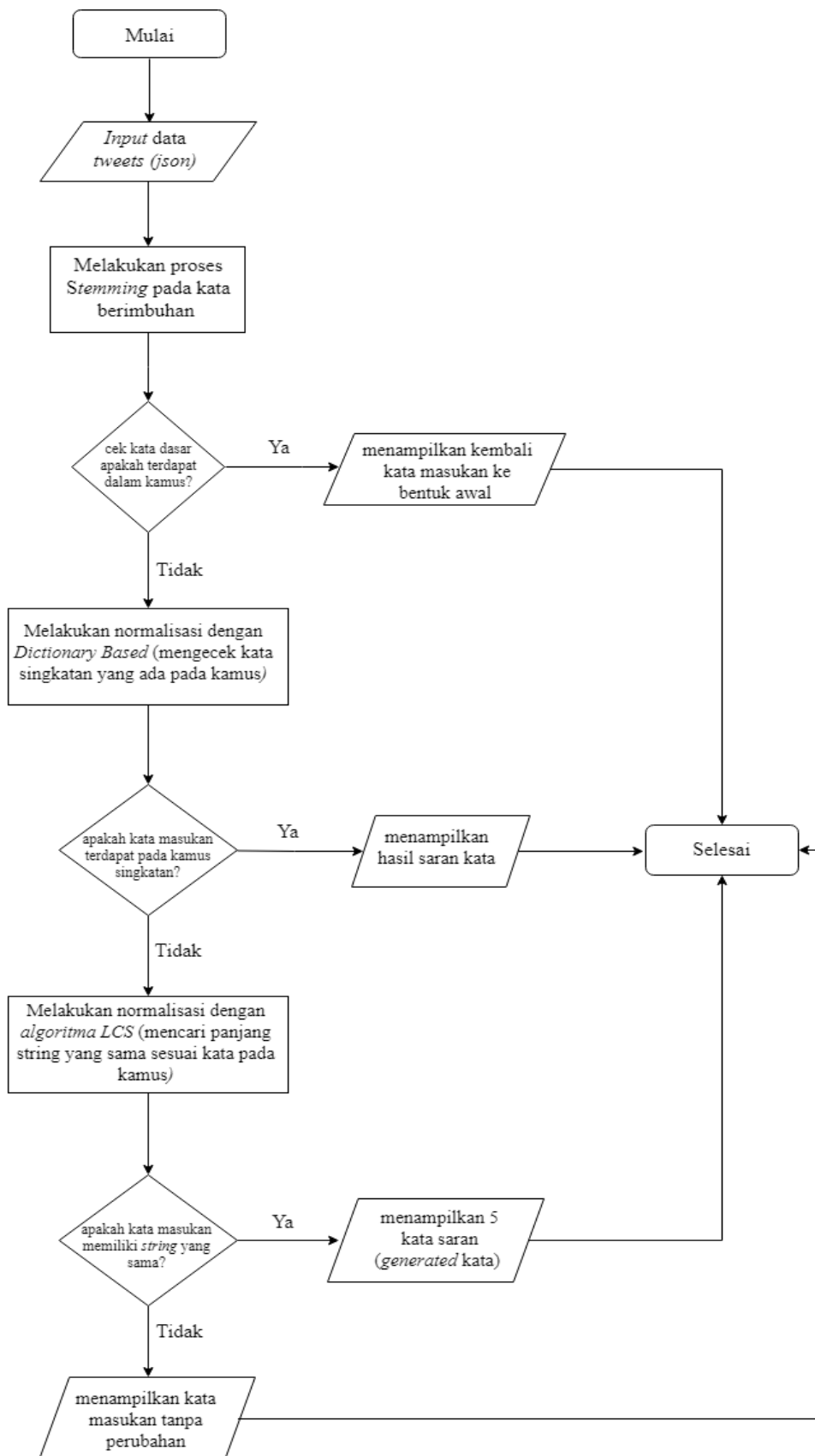
Pada aktivitas ini *user* akan melakukan *input* data yang akan dinormalisasi dan hasil normalisasi akan dapat terlihat langsung pada halaman tersebut atau bisa juga disimpan dalam *file* dengan ekstensi *json*. Diagram aktivitas normalisasi dapat dilihat pada Gambar 3.2 berikut :



Gambar 3.2. Diagram Aktivitas Sistem

3.2.3. Flowchart proses normalisasi

Menurut Pahlevy (2010) menyatakan bahwa *Flowchart* (bagan alir) merupakan sebuah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut. *Flowchart* pada penelitian ini akan menunjukkan bagaimana jalannya proses normalisasi *tweet*. *Flowchart* proses normalisasi dapat dilihat pada Gambar 3.3 berikut.



Gambar 3.3. Flowchart Proses Normalisasi

Adapun penjelasan dari *Flowchart* proses normalisasi pada Gambar 3.3. yaitu:

- a. Mulai menjalankan sistem.
- b. Melakukan *input* data *tweet* pada sistem yang berekstensi JSON. Jumlah *tweet* yang dalam satu *file* adalah 20 *tweet*.
- c. Sistem akan melakukan proses pengecekan kata. Pada proses ini akan terjadi beberapa proses yaitu, *stemming* dan normalisasi.
- d. Pada proses *stemming* sistem akan mengecek kata masukan yang mengandung imbuhan, kemudian jika sudah mendapatkan kata dasarnya maka akan dicek pada kamus. Jika kata dasar tersebut terdapat dalam kamus, maka proses *stemming* akan berhenti dan menampilkan kembali kata masukan ke bentuk semula (dengan bentuk imbuhan). Jika kata dasar tidak ditemukan, maka akan berlanjut ke proses normalisasi.
- e. Proses normalisasi akan terlebih dahulu menggunakan proses *dictionary based*, yaitu melakukan proses normalisasi dengan mengecek kata masukan pada kamus singkatan. Jika kata masukan terdapat pada kamus singkatan, maka sistem akan menampilkan hasil saran kata tersebut, tetapi jika tidak ditemukan maka proses akan berlanjut menggunakan perhitungan algoritma.
- f. Proses normalisasi menggunakan algoritma LCS (*Longest Common Subsequences*) akan mencari kesamaan antar *string* kata masukan dan kata pada kamus. Jika kata masukan memiliki *string* yang sama maka sistem akan menampilkan 5 kata saran (*word choices*) dari jumlah karakter terpendek tetapi jika tidak ditemukan kesamaan *string*, maka proses akan berhenti dengan menampilkan kata masukan tanpa perubahan.
- g. Proses normalisasi selesai.

3.2.4. Perancangan antarmuka sistem

Pada tahap perancangan sistem ini akan dilakukan perancangan menu sistem dan perancangan antarmuka dari sistem normalisasi mikroteks pada Twitter. Perancangan tampilan antarmuka pada sistem ini bertujuan untuk memberikan gambaran tampilan sistem yang akan dibangun dan juga agar pengguna dapat menjalankan aplikasi dengan mudah.

a. Rancangan halaman *upload file*

Halaman *upload file* merupakan rancangan halaman utama dari sistem ini. Pada halaman ini *user* dapat melakukan *upload file* dengan format file *json*.

The wireframe shows a web interface for uploading and normalizing tweets. At the top left is a circular placeholder labeled 'logo'. At the top right is a dropdown menu labeled 'Name'. Below the logo, the word 'Normalization' is written. The central part of the page is a large box titled 'TWEET NORMALIZER'. Inside this box, there is a section titled 'Upload a file of tweets' which contains a text input field and a 'Browse...' button.

Gambar 3.4. Rancangan halaman *upload file*

b. Rancangan halaman normalisasi

Halaman normalisasi merupakan halaman untuk menunjukkan hasil dari kalimat *tweet* yang telah dinormalisasi. Pada halaman ini terdapat perbandingan antara teks *tweet* sebelum dinormalisasi dan teks *tweet* setelah dinormalisasi. Selain itu juga terdapat *button word choices* untuk melihat *generated* kata dari hasil perhitungan sistem.

logo

Name ▼

Normalization

TWEET NORMALIZER

Upload a file of tweets

tweets
sebelum
dinormalisasi

Process Tweets

Save Result

word choices

tweets setelah
dinormalisasi

Gambar 3.5. Rancangan halaman normalisasi

Universitas Sumatera Utara

BAB 4

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan menjelaskan mengenai proses pengimplementasian *dictionary based* dan algoritma *Longest Common Subsequences* sesuai dengan perancangan sistem yang telah dilakukan pada Bab 3. Pada bab ini juga akan dilakukan pengujian sistem yang telah dibangun.

4.1. Implementasi Sistem

Sesuai dengan perancangan sistem yang telah dilakukan pada Bab 3 yaitu mulai dari tahap *pre-processing*, proses, serta metode dan algoritma yang digunakan, maka akan diimplementasikan kedalam sistem dengan menggunakan bahasa pemograman python.

4.1.1 Spesifikasi perangkat keras yang digunakan

Adapun spesifikasi perangkat keras yang digunakan untuk membangun sistem pada penelitian ini adalah sebagai berikut :

1. Processor Intel® Core™ i3-2348M CPU @ 2.30GHz 2.30GHz
2. Kapasitas Hardisk
3. Memori RAM yang digunakan 6.00 GB
4. Mouse dan keyboard

4.1.2 Spesifikasi perangkat lunak yang digunakan

Spesifikasi perangkat lunak yang digunakan untuk membangun sistem pada penelitian ini adalah sebagai berikut :

1. Sistem operasi Microsoft Windows 10 Pro Enterprise 64 bit
2. Python versi 3.7
3. JetBrains Pycharm

4.1.3. Implementasi Rancangan Antarmuka

Adapun implementasi rancangan antarmuka pada sistem yang dibangun sesuai dengan rancangan antarmuka yang telah dilakukan pada Bab 3 adalah sebagai berikut :

a. Tampilan Halaman Normalisasi

Halaman normalisasi merupakan halaman utama pada sistem yang telah dibangun. *User* dapat langsung melakukan *upload file* dan dengan menekan *button Process Tweets* proses normalisasi akan berjalan.



Gambar 4.1. Tampilan *Upload File*

b. Tampilan Hasil Normalisasi

<p>b aja sih kataku aku tdk terlalu menyukai cuma mo banyakin notif abis itu keluar tdk berperitanggung jawab hihi</p> <p>TWEET, NO. #1</p>	<p>b aja sih kataku aku tidak terlalu menyukai cuma (mob mok mol mop) banyak sedikitnya (nonaktif normatif) abis itu keluar tidak berperitanggung jawab (hati-hati hila-hila)</p> <p>NORMALIZED TWEET, NO. #1 Word Choices</p>
<p>sudhlah semua bru akan di mulai senin belajr jalani tanpa hrus berasumsi</p> <p>TWEET, NO. #2</p>	<p>sudhlah semua (baru beru biru bruk buru) akan di mulai senin belajar jalani tanpa harus berasumsi</p> <p>NORMALIZED TWEET, NO. #2 Word Choices</p>
<p>di saat politik kita begitu gaduh tak sedikit orang yg tetap bekerja dalam diam dan sepi dari tepuk tangan kades pon</p> <p>TWEET, NO. #3</p>	<p>di saat politik kita begitu gaduh tak sedikit orang yang tetap bekerja dalam diam dan sepi dari tepuk tangan kepala desa pekan olahraga nasional</p> <p>NORMALIZED TWEET, NO. #3 Word Choices</p>

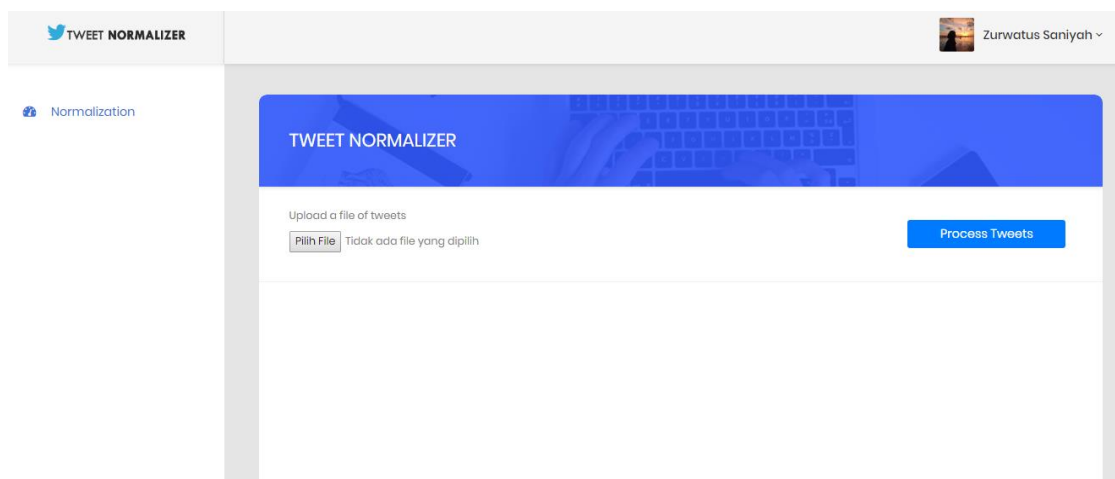
Gambar 4.2. Tampilan Hasil Normalisasi

4.1.4 Prosedur Operasional

Pada sistem normalisasi *tweet* yang menjadi halaman utama adalah halaman normalisasi yang terdapat pada Gambar 4.1. Setelah melakukan *upload file*, *user* dapat langsung menekan *button process* agar sistem dapat langsung memproses masukan yang diberikan. Proses normalisasi *tweet* akan berlangsung beberapa menit tergantung besar *file* masukan.












Prosedur operasional yang dilakukan *user* adalah sebagai berikut :

1. *Input file* dengan mengklik *button* Pilih File seperti yang ditunjukkan pada Gambar 4.3 berikut.



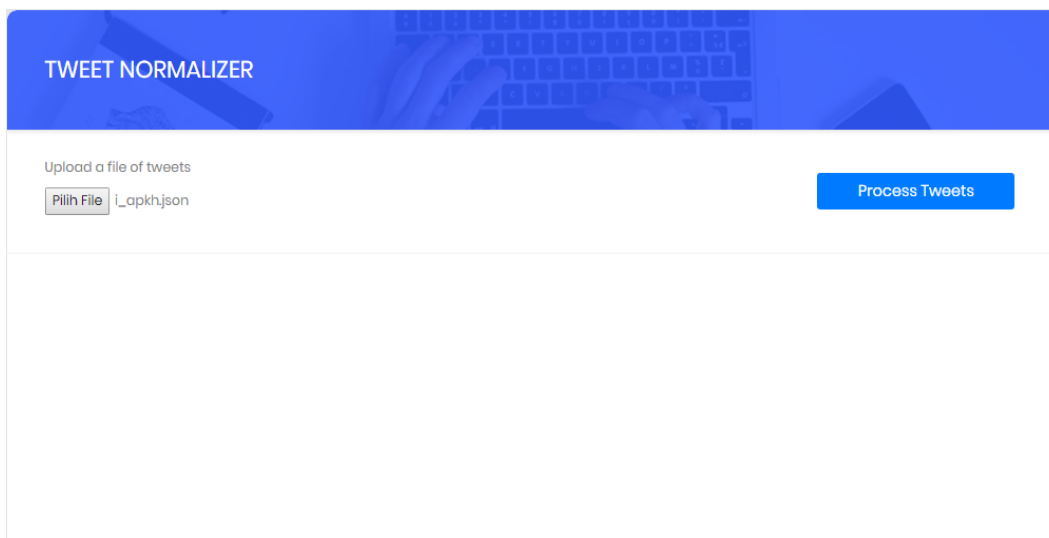
Gambar 4.3. Tampilan Pilih *File*

2. Pilih *File* yang akan diinput dengan format *json*, kemudian klik *Open* seperti yang ditunjukkan pada Gambar 4.4.

 file1.json	11/14/2018 1:55 PM	JSON File	2 KB
 file2.json	11/13/2018 2:23 PM	JSON File	3 KB
 file3.json	11/13/2018 2:28 PM	JSON File	3 KB
 file4.json	11/14/2018 1:55 PM	JSON File	3 KB
 file5.json	11/16/2018 12:18 ...	JSON File	3 KB
 file6.json	11/16/2018 1:01 PM	JSON File	3 KB
 file7.json	11/17/2018 1:16 PM	JSON File	3 KB
 file8.json	11/17/2018 2:00 PM	JSON File	3 KB
 file9.json	11/17/2018 2:46 PM	JSON File	3 KB
 file10.json	11/18/2018 3:21 PM	JSON File	3 KB
 file11.json	11/18/2018 4:40 PM	JSON File	4 KB

Gambar 4.4. Tampilan *Input File*

- Setelah *input file* yang telah dipilih, selanjutnya klik *button Process Tweets* agar sistem membaca isi *file* dan melakukan proses normalisasi



Gambar 4.5. Tampilan *Process Tweets*

- Setelah proses selesai maka sistem akan menampilkan hasil normalisasi dari *file* yang telah diinput. Pada tampilan ini dapat dilihat perbandingan dari *tweet* sebelum dinormalisasi dan setelah dinormalisasi.

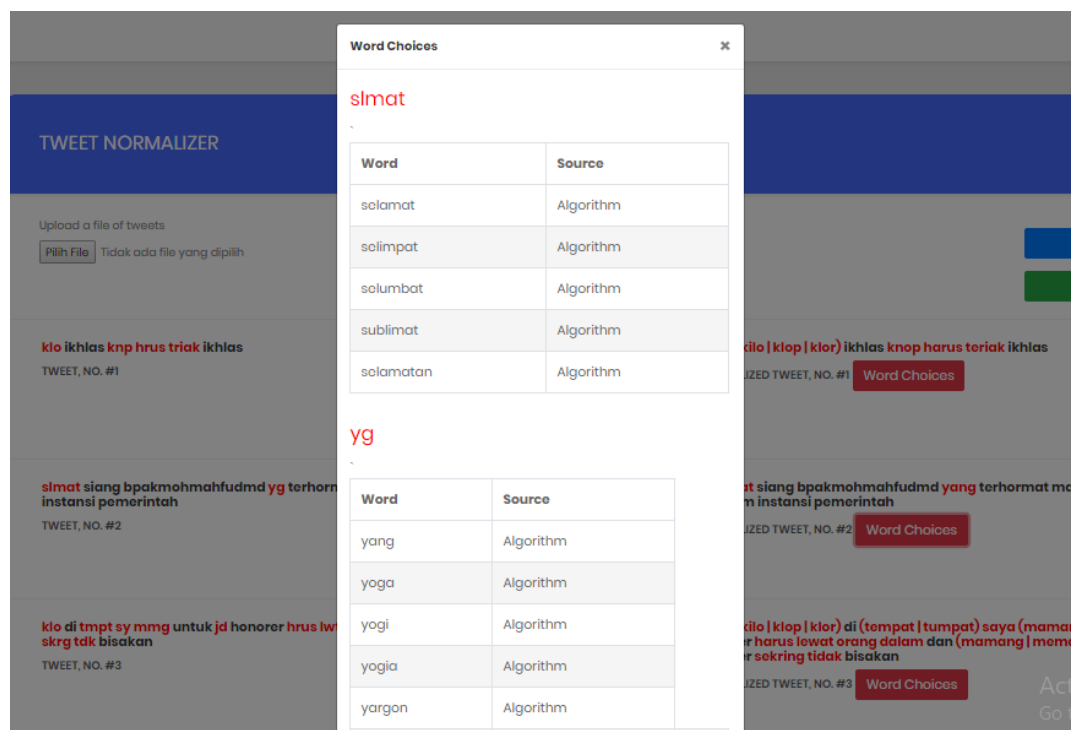
<p>b oja sih kataku aku tdk terlalu menyukai cuma mo banyakkin notif abis itu keluar tdk berperi tanggung jawab hihi</p> <p>TWEET, NO. #1</p>	<p>b oja sih kataku aku tidak terlalu menyukai cuma (mob mok mol mop) banyak sedikitnya (nonaktif normatif) abis itu keluar tidak berperi tanggung jawab (hati-hati hila-hila)</p> <p>NORMALIZED TWEET, NO. #1 Word Choices</p>
<p>sudhlah semua bru akan di mulai senin belajr jalani tanpa hrus berasumsi</p> <p>TWEET, NO. #2</p>	<p>sudhlah semua (baru beru biru bruk buru) akan di mulai senin belajar jalani tanpa harus berasumsi</p> <p>NORMALIZED TWEET, NO. #2 Word Choices</p>
<p>di saat politik kita begitu gaduh tak sedikit orang yg tetap bekerja dalam diam dan sepi dari tepuk tangan kades pon</p> <p>TWEET, NO. #3</p>	<p>di saat politik kita begitu gaduh tak sedikit orang yang tetap bekerja dalam diam dan sepi dari tepuk tangan kepala desa pekan olahraga nasional</p> <p>NORMALIZED TWEET, NO. #3 Word Choices</p>

Gambar 4.6. Tampilan Hasil Normalisasi

Keterangan :

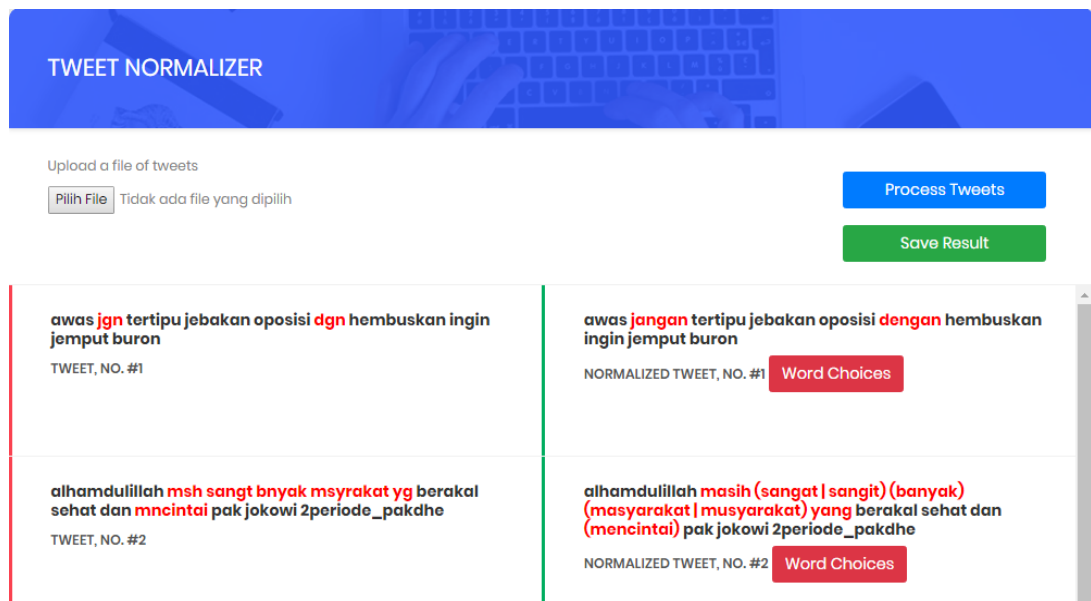
1. *Text Area 1* merupakan list *tweet* sebelum normalisasi, sedangkan *Text Area 2* merupakan list *tweet* setelah dinormalisasi.
2. Kata yang diberi warna merah merupakan kata yang mengandung singkatan, sedangkan untuk kata yang diberi warna hitam merupakan kata yang sudah benar atau tidak melewati proses normalisasi.

Pada tampilan normalisasi juga terdapat *button Word Choices* yang berguna untuk menampilkan *generated* (pilihan kata) sesuai dari perhitungan algoritma. Tampilan tersebut dapat dilihat pada Gambar 4.7.

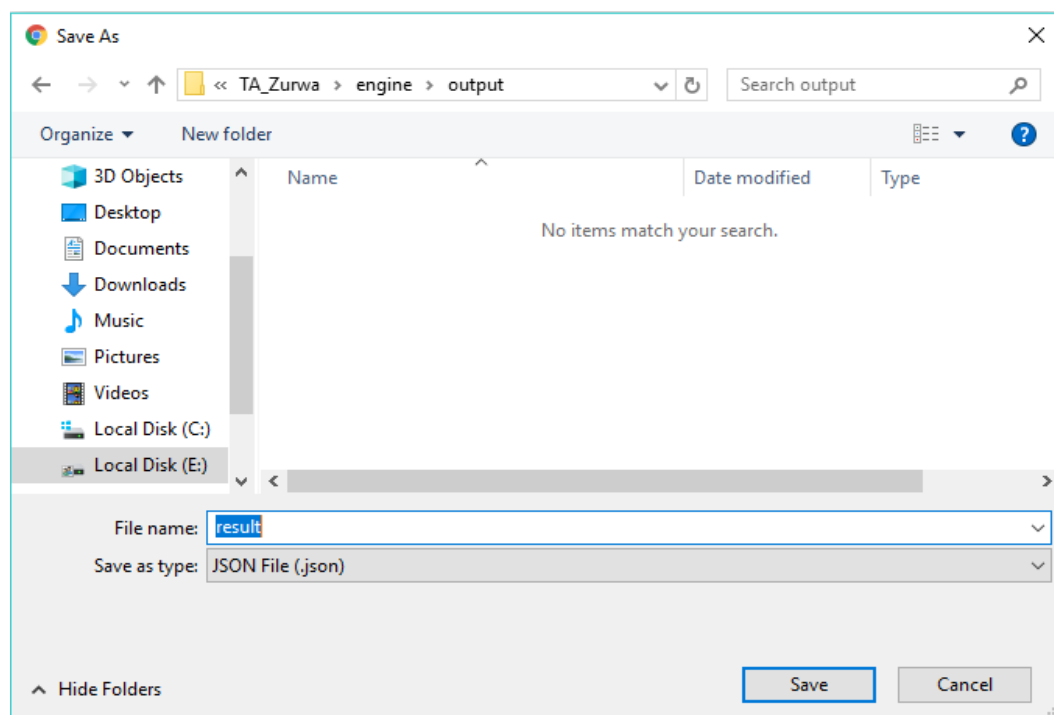


Gambar 4.7. Tampilan *Word Choices*

5. Jika *user* ingin menyimpan hasil normalisasi yang telah diproses, maka *user* dapat mengklik *button Save Result* yang ditunjukkan pada Gambar 4.8.

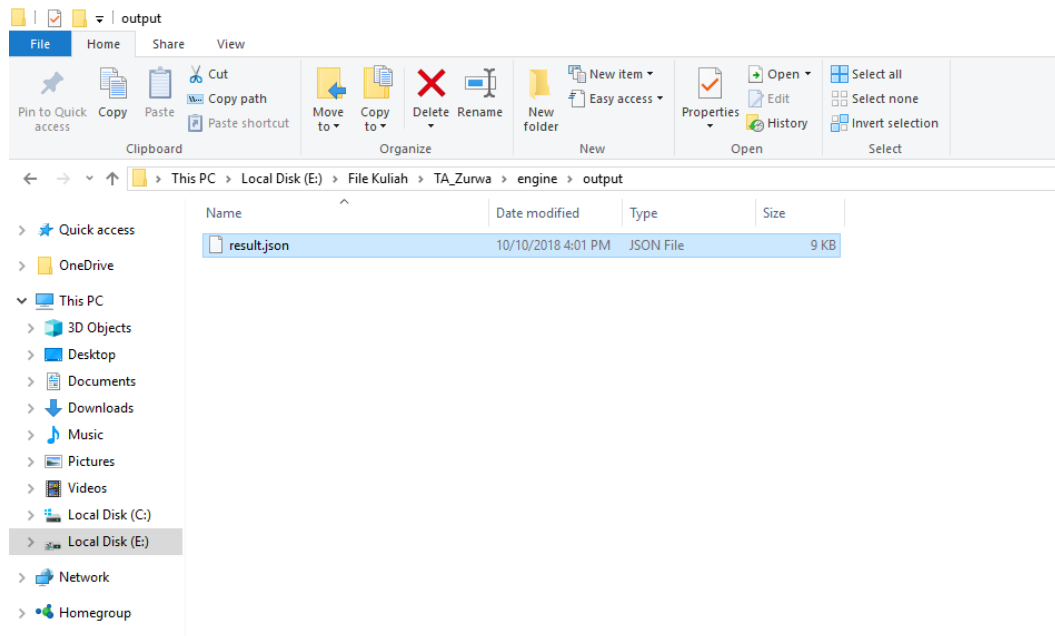


Gambar 4.8. Tampilan *Button Save Result*



Gambar 4.9. Tampilan Direktori Penyimpanan

6. Setelah selesai, maka file akan tersimpan di direktori yang telah dipilih



Gambar 4.10. File Tersimpan

4.2. Diskusi Proses Normalisasi

Diskusi proses normalisasi merupakan diskusi atau penjelasan mengenai bagaimana keputusan dalam menentukan output dari sebuah kata masukan sesuai dengan hasil implementasi sistem. Dalam diskusi ini, penulis menjabarkan proses pengecekan perkata mulai dari proses *stemming*, normalisasi menggunakan *dictionary based* dan algoritma LCS.

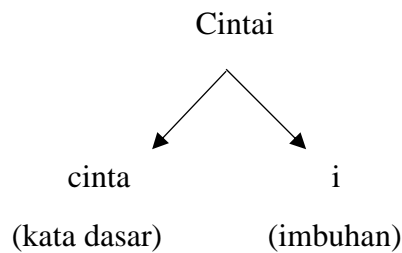
Contoh *tweet*:

[“dengerin”, “aku”, “hanya”, “mnusia”, “biasa”, “yg”, “tdk”, “mngkin”, “bsa”, “jalani”, “hidup”, “brsma”, “dia”, “selamanya”]

Dalam melakukan proses normalisasi, semua kata masukan akan dicek kebenarannya. Proses pertama yang dilakukan yaitu proses *stemming*. Seperti dijelaskan pada Bab 3, proses *stemming* diperlukan untuk mengecek kata dasar pada kata masukan yang mengandung imbuhan.

[“**dengerin**”, “aku”, “hanya”, “mnusia”, “biasa”, “yg”, “tdk”, “mngkin”, “bsa”, “**cintai**”, “dan”, “hidup”, “**brsma**”, “dia”, “selamanya”]

Pada *tweet* tersebut terdapat dua kata yang mengandung imbuhan, yaitu ‘dengerin’, ‘jalani’ dan ‘brsma’. Tetapi, pada proses pengecekan *stemming* hanya dapat dilakukan pada kata masukan yang mengandung imbuhan dan tidak merupakan singkatan atau kata baku sesuai kamus, yaitu ‘cintai’.



Setelah kata dasar diketahui, yaitu ‘cinta’ maka kata tersebut akan dicek ke kamus. Jika kata tersebut terdapat dalam kamus, maka proses *stemming* akan berhenti, karena artinya kata ‘cinta’ sudah benar sesuai kamus sehingga tidak perlu diproses dan akan dikembalikan dalam bentuk aslinya, yaitu ‘cintai’.

Selanjutnya, proses normalisasi akan berlanjut untuk mengecek kata yang mengandung singkatan. Seperti dijelaskan pada Bab 3, proses normalisasi terbagi 2, yaitu menggunakan *dictionary based* dan algoritma LCS. Proses normalisasi tersebut dapat dilihat secara lengkap pada Bab 3 bagian normalisasi.

[“dengerin”, “dr”, “dulu”, “aku”, “hanya”, “mnusia”, “biasa”, “yg”, “tdk”, “mngkin”, “bsa”, “cintai”, “dia”, “dan”, “brsma”, “selamanya”]

Dari hasil proses *dictionary based* yaitu dengan membandingkan kata masukan pada kamus singkatan dan dari proses perhitungan algoritma dengan mencari kesamaan antar *string*, maka diperoleh hasil beberapa kata saran dari proses normalisasi tersebut. Hasil kata saran dari pengecekan masing-masing kata dapat dilihat pada Tabel 4.1 berikut:

Tabel 4.1. Hasil Proses Pengecekan perkata

Kata masukan	<i>Dictionary based</i>	LCS	<i>Jumlah Karakter LCS</i>
dengerin	-	dengan sendirinya	16
dr	dana reboisasi	dar	3
		dor	3
		dur	3
		dara	4
		dari	4
dulu	-	-	-
aku	-	-	-
hanya	-	-	-
mnusia	-	manusia	7
		manusiawi	9
		mengusaikan	11
		memanusiakan	12
		mengungsikan	12
biasa	-	-	-
yg	yang	yang	4
		yoga	4
		yogi	4
		yogia	5
		yargon	6
tdk	tidak	tedak	5
		tidak	5
		todak	5
		tudak	5
		tandak	6
mngkin	-	mangkin	7
		mungkin	7
		mengking	8
		melengking	10
		menengking	10

Tabel 4.1. Hasil Proses Pengecekan perkata (Lanjutan)

bsa	-	basa	4
		bisa	4
		busa	4
		basah	5
		basal	5
cintai	-	-	-
dia	-	-	-
dan	-	-	-
brsma	-	bersama	7
		bersemak	8
		biprisma	8
		berasmara	9
		berasrama	9
selamanya	-	-	-

Pada Tabel 4.1 tersebut telah dijabarkan pengecekan perkata sesuai dengan *output* yang dihasilkan dari *dictionary based* dan algoritma LCS. Beberapa kata masukan menghasilkan *output* dari *dictionary based* dan algoritma LCS, seperti ‘dr’, ‘yg’ dan ‘tdk’. Tetapi, karena dalam penelitian ini lebih mengutamakan hasil dari *dictionary based*, maka *output* yang akan ditampilkan, yaitu ‘dana reboisasi’, ‘yang’ dan ‘tidak’. Kata masukan ‘dr’ sebenarnya jika dicek secara manual dengan mempertimbangkan struktur antar kalimat, maksud dari kata tersebut adalah ‘dari’ yang didapat dari perhitungan LCS dengan jumlah karakter terpendek. Tetapi, karna pada kamus singkatan terdapat singkatan ‘dr’ yaitu ‘dana reboisasi’, maka *output* yang dikeluarkan adalah ‘dana reboisasi’. *Output* dari hasil perhitungan *dictionary based*, yaitu:

[“dengerin”, “**dana reboisasi**”, “dulu”, “aku”, “hanya”, “mnusia”, “biasa”, “**yang**”, “**tidak**”, “mngkin”, “bsa”, “**cintai**”, “dia”, “dan”, “brsma”, “selamanya”]

Selanjutnya, kata masukan yang tidak dihasilkan dari *dictionary based* tetapi hanya dihasilkan hanya dari perhitungan LCS, yaitu ‘dengerin’, ‘mnusia’, ‘mngkin’, ‘bsa’, dan ‘brsma’.

-dengerin

Kata ‘dengerin’ menghasilkan *output* ‘dengan sendirinya’, karena dalam perhitungan LCS melihat kedekatan antar string seperti dijelaskan pada Bab 3. Perbandingan antara *string* kata masukan ‘dengerin’ dan *string* kata *output* ‘dengan sendirinya’ dapat dilihat pada perbandingan S1 dan S2.

Perbandingan S1 dan S2:

S1 = D E N G E R I N

S2 = D E N G A N S E N D I R I N Y A

common subsequence = D E N G E R I N

Keterangan : *cell* yang dihitamkan untuk urutan pertama menandakan karakter yang masuk dalam LCS, yaitu ‘d’, ‘e’, ‘n’, ‘g’, ‘e’, ‘r’, ‘i’, dan ‘n’.

-mnusia

Kata masukan ‘mnusia’ menghasilkan beberapa kata saran dari perhitungan LCS. Perhitungan jumlah karakter diperlukan untuk melihat perbandingan dari kata saran. Persamaan untuk menghitung jumlah karakter setiap kata saran dapat dilihat pada Bab 3.

mnusia	manusia	7
	manusiawi	9
	mengusaikan	11
	memanusiakan	12
	mengungsikan	12

Dengan melihat perhitungan jumlah karakter, karakter terpendek diperoleh dari kata ‘manusia’. Oleh karena itu, *output* yang dihasilkan yaitu ‘manusia’.

[“dengan sendirinya”, “dana reboisasi”, “dulu”, “aku”, “hanya”, “manusia”, “biasa”, “yang”, “tidak”, “mngkin”, “bsa”, “cintai”, “dia”, “dan”, “brsma”, “selamanya”]

-mngkin

Kata masukan ‘mngkin’ menghasilkan beberapa saran kata yang dihasilkan dari perhitungan LCS, yaitu:

mngkin	mangkin	7
	mungkin	7
	mengking	8
	melengking	10
	menengking	10

Dengan melihat perhitungan jumlah karakter terpendek diperoleh dua kata saran, yaitu ‘mangkin’ dan ‘mungkin’ dengan jumlah karakter 7.

[“dengan sendirinya”, “dana reboisasi”, “dulu”, “aku”, “hanya”, “manusia”, “biasa”, “yang”, “tidak”, (“mangkin” | “mungkin”), “bsa”, “cintai”, “dia”, “dan”, “brsma”, “selamanya”]

-bsa

Kata masukan ‘bsa’ menghasilkan beberapa saran kata yang dihasilkan dari perhitungan LCS, yaitu:

bsa	basa	4
	bisa	4
	busa	4
	basah	5
	basah	5

Dengan melihat perhitungan jumlah karakter terpendek diperoleh tiga kata saran, yaitu ‘basa’, ‘bisa’, dan ‘busa’ dengan jumlah karakter 4 . Oleh karena itu, *output* yang ditampilkan memiliki tiga saran kata.

[“dengan sendirinya”, “dana reboisasi”, “dulu”, “aku”, “hanya”, “manusia”, “biasa”, “yang”, “tidak”, (“mangkin” | “mungkin”), (“basa” | “bisa” | “busa”), “cintai”, “dia”, “dan”, “brsma”, “selamanya”]

-brsma

Kata masukan ‘brsma’ menghasilkan beberapa saran kata dari perhitungan LCS, yaitu:

brsma	bersama	7
	bersemak	8
	biprisma	8
	berasmara	9
	berasrama	9

Jumlah karakter terpendek hanya dihasilkan oleh kata saran ‘bersama’ dengan jumlah karakter 7. Maka *output* yang ditampilkan:

[“dengan sendirinya”, “dana reboisasi”, “dulu”, “aku”, “hanya”, “manusia”, “biasa”, “yang”, “tidak”, (“mangkin” | “mungkin”), (“basa” | “bisa” | “busa”), “cintai”, “dia”, “dan”, “bersama”, “selamanya”]

Dari hasil pengecekan perkata, maka dapat dilihat perbandingan antara *tweet* sebelum dinormalisasi dengan *tweet* setelah dinormalisasi dari Tabel 4.2 berikut:

Tabel 4.2. Perbandingan *Tweet*

<i>Tweet</i> sebelum dinormalisasi	<i>Tweet</i> setelah dinormalisasi
[“dengerin dr dulu aku hanya mnusia biasa yg tdk mngkin bsa cintai dia dan brsma selamanya”]	[“dengan sendirinya dana reboisasi dulu aku hanya manusia biasa yang tidak (mangkin mungkin) (basa bisa busa) cintai dia dan bersama selamanya”]

4.3. Pengujian Sistem

Setelah implementasi sistem selesai, maka akan dilakukan pengujian dari sistem yang telah dibangun. Pengujian sistem bertujuan untuk mengukur seberapa besar tingkat akurasi yang didapat dengan menggunakan algoritma *Longest Common Subsequences* dan juga mengetahui apakah sistem bekerja dengan baik atau tidak.

Dalam menghitung tingkat akurasi, digunakan perhitungan normalisasi secara manual dengan menghitung menentukan TP, TN, FN, dan FP.

a. TP (*True Positive*)

True positive yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem. Contohnya kata masukan ‘tidak’ tidak dikoreksi oleh sistem atau tidak mengalami perubahan, tetap pada bentuk awal yaitu ‘tidak’.

b. TN (*True Negative*)

True negative yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem. Contohnya kata masukan ‘yg’ dinormalisasi oleh sistem menjadi ‘yang’.

c. FN (*False Negative*)

False negative yaitu jumlah data negatif namun terklasifikasi salah oleh sistem. Contohnya yaitu kata ‘dr’ dinormalisasi oleh sistem menjadi ‘dana reboisasi’, sedangkan jika dikoreksi secara manual maksud dari kata tersebut adalah ‘dari’.

d. FP (*False Positive*)

False positive yaitu jumlah data positif namun terklasifikasi salah oleh sistem. Contohnya yaitu kata masukan ‘dengerin’ mengalami perubahan menjadi ‘dengan sendirinya’, sedangkan jika dikoreksi secara manual kata tersebut sebenarnya sudah benar, tetapi karna kata tersebut tidak merupakan kata baku maka algoritma mencari kedekatan kata tersebut dengan kata pada kamus.

Pengujian TP, TN, FN, dan FP akan dilakukan dengan mengecek kata perkata dari sebuah *tweet* secara manual. Jumlah dataset yang diuji berjumlah 20 dataset dengan 400 *tweets*. Salah satu *tweet* yang akan diuji dapat dilihat pada Tabel 4.3.

Contoh *tweet* :

[“jgn”, “banyakin”, “masalah”, “cobalah”, “trima”, “dgn”, “ikhlas”, “dr”, “hti”]

Tabel 4.3. Pengujian Sistem

Kata masukan	Hasil Normalisasi	Keterangan
jgn	jangan	TN
banyakin	banyak sedikitnya	FP
masalah	masalah	TP
cobalah	cobalah	TP
	basa	
bsa	bisa	TN
	busa	

Tabel 4.3. Pengujian Sistem (Lanjutan)

trima	terima	TN
dgn	dengan	TN
ikhlas	ikhlas	TP
dr	dana reboisasi	FN
hti	hati	TN

Hasil pengujian dari *tweet* tersebut yaitu dengan jumlah TP = 3, TN = 5, FN = 1, dan FP = 1. Hasil pengujian dari dataset secara keseluruhan akan ditampilkan pada tiga bagian, yaitu hasil pengujian menggunakan algoritma LCS, hasil pengujian menggunakan *dictionary based*, dan hasil pengujian gabungan, yaitu dengan menggunakan algoritma LCS dan *dictionary based*. Adapun hasil pengujian sistem tersebut ditunjukkan pada masing-masing tabel, yaitu Tabel 4.4, Tabel 4.5, dan Tabel 4.6.

Tabel 4.4. Hasil Pengujian Sistem Menggunakan Algoritma LCS

Dataset (<i>json</i>)	Jumlah <i>Tweet</i> Dalam File	Hasil Normalisasi			
		TP	TN	FN	FP
<i>dataset1.json</i>	20	200	34	75	3
<i>dataset2.json</i>	20	232	15	62	8
<i>dataset3.json</i>	20	227	26	33	11
<i>dataset4.json</i>	20	192	30	67	20
<i>dataset5.json</i>	20	211	15	44	22
<i>dataset6.json</i>	20	222	10	38	29
<i>dataset7.json</i>	20	157	29	65	18
<i>dataset8.json</i>	20	164	42	59	15
<i>dataset9.json</i>	20	184	26	65	33
<i>dataset10.json</i>	20	205	40	76	24
<i>dataset11.json</i>	20	127	29	40	24
<i>dataset12.json</i>	20	146	20	44	13
<i>dataset13.json</i>	20	175	31	35	18
<i>dataset14.json</i>	20	151	31	30	27

Tabel 4.4. Hasil Pengujian Sistem Menggunakan Algoritma LCS (Lanjutan)

<i>dataset15.json</i>	20	129	20	28	15
<i>dataset16.json</i>	20	207	42	50	16
<i>dataset17.json</i>	20	157	35	59	10
<i>dataset18.json</i>	20	154	33	46	30
<i>dataset19.json</i>	20	179	20	62	19
<i>dataset20.json</i>	20	160	38	70	20
Total	400	3579	566	1048	375

Berdasarkan hasil pengujian pada Tabel 4.4, dapat diketahui jumlah $TP = 3579$, $TN = 566$, $FN = 1048$, dan $FP = 375$ dari 400 *tweet*. Berdasarkan hasil pengujian yang telah dilakukan, akan dilakukan perhitungan untuk mendapatkan nilai akurasi, presisi, recall, dan *f-score*.

a. Akurasi

Akurasi merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual. Adapun persamaan untuk menghitung tingkat akurasi yaitu dengan persamaan (1) berikut.

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP + TN}{TP + TN + FN + FP} \times 100\% \\
 &= \frac{3579 + 566}{3579 + 566 + 1048 + 375} \times 100\% \\
 &= 0.74 \times 100\% = \mathbf{74\%}
 \end{aligned}
 \tag{1}$$

b. Presisi

Presisi merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Persamaan untuk menghitung tingkat presisi yaitu dengan persamaan (2) berikut.

$$\text{Presisi} = \frac{TP}{FP + TP} \times 100\%
 \tag{2}$$

$$\begin{aligned}
 &= \frac{3579}{375 + 3579} \times 100\% \\
 &= 0.90 \times 100\% = \mathbf{90\%}
 \end{aligned}$$

c. Recall

Recall merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Persamaan untuk menghitung recall yaitu dengan persamaan (3) berikut.

$$\begin{aligned}
 \mathbf{Recall} &= \frac{TP}{FN + TP} \times 100\% \quad (3) \\
 &= \frac{3579}{1048 + 3579} \times 100\% \\
 &= 0.77 \times 100\% = \mathbf{77\%}
 \end{aligned}$$

d. F-Measure (F_1 - Score)

F-Measure (F_1 - Score) yaitu pengukuran yang mengkombinasikan presisi dan recall yang diterapkan ke dalam deret harmonik. Range untuk nilai F-Measure adalah antara 0-1. Persamaan untuk menghitung F-Measure yaitu dengan persamaan (4) berikut.

$$\begin{aligned}
 F_1\text{-Score} &= 2 \times \frac{\text{presisi} \times \text{recall}}{\text{presisi} + \text{recall}} \quad (4) \\
 &= 2 \times \frac{0.90 \times 0.77}{0.90 + 0.77} \\
 &= 2 \times \frac{0.693}{1.67} = \mathbf{0.82}
 \end{aligned}$$

Tabel 4.5. Hasil Pengujian Sistem Menggunakan *Dictionary Based*

Dataset (json)	Jumlah Tweet Dalam File	Hasil Normalisasi			
		TP	TN	FN	FP
<i>dataset1.json</i>	20	203	48	60	1
<i>dataset2.json</i>	20	211	47	59	0
<i>dataset3.json</i>	20	233	28	36	0
<i>dataset4.json</i>	20	197	51	59	2
<i>dataset5.json</i>	20	245	15	30	2
<i>dataset6.json</i>	20	239	22	35	3
<i>dataset7.json</i>	20	179	24	61	5
<i>dataset8.json</i>	20	200	29	50	1
<i>dataset9.json</i>	20	211	30	59	8
<i>dataset10.json</i>	20	238	33	72	2
<i>dataset11.json</i>	20	153	21	46	0
<i>dataset12.json</i>	20	152	26	42	3
<i>dataset13.json</i>	20	190	24	42	3
<i>dataset14.json</i>	20	173	25	40	1
<i>dataset15.json</i>	20	135	20	34	3
<i>dataset16.json</i>	20	252	18	40	5
<i>dataset17.json</i>	20	173	31	52	5
<i>dataset18.json</i>	20	184	30	46	3
<i>dataset19.json</i>	20	186	35	58	1
<i>dataset20.json</i>	20	233	30	20	5
Total	400	3987	587	941	53

Berdasarkan hasil pengujian pada Tabel 4.5, dapat diketahui jumlah TP = 3987, TN = 587, FN = 941, dan FP = 53 dari 400 *tweet* dan akan dilakukan perhitungan akurasi, presisi, recall, dan *f-score* seperti persamaan yang telah dijelaskan pada hasil pengujian sebelumnya.

a. Akurasi

$$\begin{aligned} \textbf{Akurasi} &= \frac{3987 + 587}{3987 + 587 + 941 + 53} \times 100\% \\ &= 0.82 \times 100\% = \mathbf{82\%} \end{aligned}$$

b. Presisi

$$\begin{aligned} \textbf{Presisi} &= \frac{3987}{53 + 3987} \times 100\% \\ &= 0.98 \times 100\% = \mathbf{98\%} \end{aligned}$$

c. Recall

$$\begin{aligned} \textbf{Recall} &= \frac{3987}{941 + 3987} \times 100\% \\ &= 0.80 \times 100\% = \mathbf{80\%} \end{aligned}$$

d. F-Measure (F_1 - Score)

$$\begin{aligned} \textbf{f - score} &= 2 \times \frac{0.98 \times 0.80}{0.98 + 0.80} \\ &= 2 \times \frac{0.784}{1.78} = \mathbf{0.88} \end{aligned}$$

Tabel 4.6. Hasil Pengujian Sistem Menggunakan LCS dan *Dictionary Based*

Dataset (json)	Jumlah Tweet Dalam File	Hasil Normalisasi			
		TP	TN	FN	FP
<i>dataset1.json</i>	20	196	80	32	4
<i>dataset2.json</i>	20	231	62	19	5
<i>dataset3.json</i>	20	225	49	18	5
<i>dataset4.json</i>	20	181	74	33	21
<i>dataset5.json</i>	20	189	45	35	23
<i>dataset6.json</i>	20	222	28	21	28
<i>dataset7.json</i>	20	166	52	38	13
<i>dataset8.json</i>	20	170	65	28	17

Tabel 4.6. Hasil Pengujian Sistem Menggunakan LCS dan *Dictionary Based*
(Lanjutan)

<i>dataset9.json</i>	20	186	51	50	21
<i>dataset10.json</i>	20	218	70	40	17
<i>dataset11.json</i>	20	123	59	18	20
<i>dataset12.json</i>	20	144	44	20	15
<i>dataset13.json</i>	20	175	38	24	22
<i>dataset14.json</i>	20	155	32	21	31
<i>dataset15.json</i>	20	125	29	19	19
<i>dataset16.json</i>	20	240	43	13	19
<i>dataset17.json</i>	20	163	49	39	10
<i>dataset18.json</i>	20	160	44	26	33
<i>dataset19.json</i>	20	198	32	28	22
<i>dataset20.json</i>	20	215	38	14	21
Total	400	3682	984	536	366

Berdasarkan hasil pengujian pada Tabel 4.6, dapat diketahui jumlah TP = 3682, TN = 984, FN = 536, dan FP = 366 dari 400 *tweet*. Berdasarkan hasil pengujian yang telah dilakukan, akan dilakukan perhitungan untuk mendapatkan nilai akurasi, presisi, recall, dan *f-score* menggunakan persamaan seperti perhitungan pada pengujian sebelumnya.

a. Akurasi

$$\begin{aligned}
 \text{Akurasi} &= \frac{3682 + 984}{3682 + 984 + 536 + 366} \times 100\% \\
 &= 0.83 \times 100\% = \mathbf{83\%}
 \end{aligned}$$

b. Presisi

$$\begin{aligned}
 \text{Presisi} &= \frac{3682}{366 + 3682} \times 100\% \\
 &= 0.90 \times 100\% = \mathbf{90\%}
 \end{aligned}$$

c. Recall

$$\begin{aligned}
 \text{Recall} &= \frac{3682}{536 + 3682} \times 100\% \\
 &= 0.87 \times 100\% = \mathbf{87\%}
 \end{aligned}$$

d. F-Measure (F_1 - Score)

$$\begin{aligned}
 f - score &= 2 \times \frac{0.90 \times 0.87}{0.90 + 0.87} \\
 &= 2 \times \frac{0.78}{1.77} = \mathbf{0.88}
 \end{aligned}$$

Berdasarkan hasil perhitungan dari masing-masing pengujian, tingkat akurasi, presisi, recall, dan *f-score* yang dihasilkan berbeda-beda. Hasil perbandingan dari masing-masing pengujian akan terlihat lebih jelas pada Tabel 4.7 berikut.

Tabel 4.7. Perbandingan Hasil Perhitungan Akurasi, Presisi, Recall, dan *F-Score*

	Hasil Perhitungan			
	Akurasi	Presisi	Recall	<i>F-Score</i>
LCS	74%	90%	77%	0.82
<i>Dictionary Based</i>	82%	98%	80%	0.88
LCS dan <i>Dictionary Based</i>	83%	90%	87%	0.88

Berdasarkan hasil perbandingan pada Tabel 4.7 tersebut, terlihat bahwa hasil perhitungan dengan menggabungkan algoritma LCS dan *dictionary based* akan memberikan hasil yang lebih baik dibandingkan jika hanya menggunakan LCS tersendiri atau metode *dictionary based* tersendiri. Pada tabel tersebut, hasil dengan menggunakan *dictionary based* sedikit lebih baik dibandingkan LCS. Oleh karena itu, dalam penelitian ini *dictionary based* sangat membantu dalam proses normalisasi untuk memaksimalkan hasil pengujian.

BAB 5

KESIMPULAN DAN SARAN

Pada bab ini membahas mengenai kesimpulan yang dapat diperoleh dari hasil penelitian yang telah diimplementasikan dan juga saran-saran yang dapat digunakan untuk penelitian atau pengembangan sistem ini selanjutnya.

5.1. Kesimpulan

Dari hasil pengujian sistem pada Bab 4, dapat diambil kesimpulan sebagai berikut :

1. Algoritma *Longest Common Subsequences* dapat diterapkan untuk normalisasi teks Twitter berbahasa Indonesia dengan tingkat akurasi 83%, presisi 90%, recall 87%, dan F1-Score 0.88 dengan jumlah data uji sebanyak 400 *tweet*.
2. *Dictionary based* yang diimplementasikan dapat membantu dalam proses normalisasi dan memaksimalkan hasil normalisasi.

5.2. Saran

Adapun saran yang dapat diberikan oleh penulis untuk pengembangan penelitian selanjutnya adalah sebagai berikut :

1. Menggunakan korpus tambahan, seperti korpus kata gaul atau slang yang biasa digunakan sehari-hari agar lebih memaksimalkan perbaikan kata.
2. Penambahan pendeteksian bahasa asing agar bahasa asing tidak ikut terkoreksi.
3. Menerapkan algoritma lain untuk mendapatkan hasil normalisasi yang maksimal dan tingkat akurasi yang lebih tinggi

DAFTAR PUSTAKA

- Aziz. 2013. Sistem pengklasifikasian entitas pada pesan Twitter menggunakan ekspresi regular dan *naïve bayes* [skripsi]. Bogor (ID). Institut Pertanian Bogor.
- Design and Analysis of Algorithms. 2015. cs.cmu.edu, 28 Januari 2015 (diakses 14 Maret 2018).
- Dutoit, T. 1997. An Introduction to Text-to-Speech Synthesis. Kluwer Academic Publisher. Dordrecht 1997.
- Hanafiah, N., Kevin, A., Sutanto, C., Arifin, Y., & Hartanto, J. 2017. Text Normalization Algorithm on Twitter in Complaint Category. 2nd International Conference on Computer Science and Computational Intelligence 2017 (ICCSCI), pp. 20-26.
- Irawan, Y. 2016. Normalisasi Teks Twitter Bahasa Indonesia Berbasis *Noisy Channel Model* [skripsi]. Yogyakarta[ID]. Universitas Gajah Mada.
- Khoury, R. 2015. Microtext Normalization using Probably-Phonetically-Similar Word Discovery. Canada. Lakehead University.
- Manning, C.D., Raghavan, P., & Schütze, H. (2009). An Introduction to Information Retrieval. Cambridge: Cambridge University Press. (Online) <http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf> (1 Oktober 2018).
- Pahlevy, 2010. Pengertian Flowchart dan definisi data. (Online) <http://www.landasanteori.com/2015/10/pengertian-flowchart-dan-defenisi-data.html> (20 Agustus 2018).
- Putranto, D.D. 2008. Pengkajian Masalah Longest Common Subsequences [jurnal]. Bandung[ID]. Institut Teknologi Bandung.
- Saragih, Tri. 2017. Normalisasi Teks Pada Teks Twitter Berbahasa Indonesia Menggunakan Algoritme Jarak String Pada R [skripsi]. Bogor (ID). Institut Pertanian Bogor.

- Sarwani MZ., Mahmudy WY. 2015. Analisis Twitter untuk mengetahui karakter seseorang menggunakan algoritma naïve bayes classifier. Seminar Nasional Sistem Informasi Indonesia (SESINDO). 2015 Nov 2-3; Surabaya, Indonesia. Malang (ID).
- Satapathy, R., Guerreiro, C., Chaturvedi, I., & Cambria, E. 2017. Phonetic-Based Microtext Normalization for Twitter Sentiment Analysis. In 2017 *IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 407-413). IEEE.
- Sugiarti, Y. 2013. Analisis dan Perancangan UML Generated VB6. Yogyakarta. Graha Ilmu.
- Wahyuningtyas, A. 2016. Deteksi spam pada Twitter menggunakan algoritme naïve bayes [skripsi]. Bogor (ID). Institut Pertanian Bogor.
- Xue, Z., Yin, D., & Davison, B. (2011). Normalizing Microtext. Department of Computer Science & Engineering, Lehigh University Bethlehem, PA 18015 USA.

LAMPIRAN

Lampiran 1

Contoh Kateglo entri kamus

Kata		
-an	A	Abangan
-anda	Ab	Abangga
-asi	ab-	Abar
-da	Aba	abar-abar
-el	aba-aba	abar-abar pintu
-em	Abad	Abaran
-er-	abad keemasan	Abatoar
-i	abad komputer	Abau
-iah	abad modern	Abdas
-if	abad pertengahan	Abdi
-is	Abadi	abdi dalem
-isasi	abadiah	abdi masyarakat
-isme	Abah	abdi negara
-itas	abah-abah	Abdikasi
-kah	abah-abah kuda	Abdomen
-lah	abah-abah perahu	Abdominal
-man	abah-abah tenun	Abdu
-mu	Abai	Abduksi
-nda	Abaian	Abductor
-ni	abaimana	Abdul
-nya	Abaka	Abece
-wan	abaktinal	Aben
-wati	abakus	Aberasi
-wi	abal-abal	aberasi cahaya
-wiah	Aban	aberasi kromosom
a	Abang	Abet

Lampiran 2

Contoh Kateglo entri singkatan

Kata		Kata	
Singkatan	Kepanjangan	Singkatan	Kepanjangan
a.d.	atas dasar	capeg	calon pegawai
a.i.	ad interim	cergam	cerita bergambar
a.l.	antara lain	cerpen	cerita pendek
a.n.	atas nama	ckp	Cukup
AAL	Akademi Angkatan Laut	curhat	curahan hati
ABG	anak baru gede	DBD	demam berdarah dengue
Abrip	Ajun Brigadir Polisi	dg	dengan
Angkuta	Angkutan Kota	Dikti	Pendidikan Tinggi
angmor	angkutan bermotor	Dir	direktur
asbun	asal bunyi	dkk.	dan kawankawan
ASI	air susu ibu	dl	dulu
askes	asuransi kesehatan	dll.	dan lainlain
Babel	Bangka dan Belitung	dlm	dalam
baleg	badan legislasi	dmn	di mana
bandara	bandar udara	DNI	Daftar Negatif Investasi
banpol	bantuan polisi	dpt	dapat
banser	Barisan Ansor Serbaguna	DR	Dana Reboisasi
bemo	becak bermotor	dr.	dokter
BIDSUS	Bidang Khusus	DRN	Dewan Riset Nasional
bkn	bukan	dsb.	dan sebagainya
blm	belum	dst.	dan seterusnya
bls	balas	dtg	dating
BM	bahasa Melayu	DTK	Dinas Tata Kota
BNI	Bank Negara Indonesia	Dubes	duta besar
brp	berapa	EGP	emangnya gua pikirin
bs	bisa	fax	faksimile
BSF	Badan Sensor Film	ga	nggak

Lampiran 3

Contoh data *tweet* mengandung singkatan atau kata tidak baku

<i>Tweet</i>
awas jgn tertipu jebakan oposisi dgn hembuskan ingin jemput buron
alhamdulillah msh sangt bnjak msyrakat yg berakal sehat dan mncintai pak 63ppara 2periode_pakdhe
duit tak mngizinkan 3
barangkali disana ada jawabnya mngapa ditanahku terjadi bencana mngkin tuhan mulai bosan melihat tingkah kita
tergantung dilihat dr sudut kamera mana bs dilihat jg sbg teguran nasihat saran lagipula
haiiyyaaa 7 buah rumah tdk layak huni di kp guha desa ciminyak kecmuncang kablebak sdh slsai di rehab
bahkan jk hidup terasa sangat menyakitkan dan terasa susah seseorang itu harusnya merasa bersyukur bhwa mereka tetap hdp yato noragami
rkyt gregetan 63pparat sdh sering bahkan byk terlht ngga adil kl kubu oposisi atw kontra rezi
rezeki yg sebenar itu ialah yg x nmpk pd mata tp terasa pd hati ada org naik motor kerusi cuma muat 2 org anak kecil dl
kalo lg gini kgm shbt gua nih yang belabelain dengerin gua nangis ditelfon bawain makanan kkosan ngobrol gajelas
b aja sih kataku aku tdk terlalu menyukai cuma mo banyakkin notif abis itu keluar tdk berperilaku tanggung jawab hihi
sudhlah semua bru akan di mulai senin belajr jalani tanpa hrus berasumsi
akhirnya dia mnikah juga slmt pengantin baru

Lampiran 4

Data Pendukung Asumsi

No.	Singkatan	Hasil LCS	Jumlah Karakter	Keterangan
1	mngapa	mengapa	7	Benar
		mengapai	8	Salah
		mengapam	8	Salah
2	hrus	harus	5	Benar
		humerus	7	Salah
		hidraulis	9	Salah
3	triak	teriak	6	Benar
		teriakan	8	Salah
		terinjak	8	Salah
4	sblm	sublim	6	Salah
		sebelum	7	Benar
		sublema	7	Salah
5	sy	saya	4	Benar
		sayu	4	Salah
		syah	4	Salah
6	tlah	talah	5	Salah
		telah	5	Benar
		tulah	5	Salah
		telaah	6	Salah
7	bgitu	begitu	6	Benar
		begituan	8	Salah
		begitu pun	9	Salah
8	byran	bayaran	7	Benar
		berkebyaran	11	Salah
		berkeluyuran	12	Salah
9	mkn	makan	5	Benar
		makin	5	Salah
		makna	5	Salah
10	utk	utik	4	Salah
		untuk	6	Benar
		utrikel	7	Benar
11	tdk	tedak	5	Benar
		tidak	5	Salah
		todak	5	Salah
12	mnikah	menikah	7	Benar
		menikahi	8	Salah
		meningkah	9	Salah
13	slh	slah	4	Salah
		salah	5	Benar
		saleh	5	Salah
14	sgt	sangat	6	Benar
		sangit	6	Salah
		sangat	7	Salah

15	byk	banyak	6	Benar
		banyakan	8	Salah
		berbanyak	9	Salah
16	msyrakat	masyarakat	10	Benar
		musyarakat	10	Salah
		memasyarakat	13	Salah
17	slsi	selesai	7	Benar
		selesaian	9	Salah
		salinisasi	11	Salah
18	rkyt	rakyat	6	Benar
		rukyyat	6	Salah
		rakyat biasa	11	Salah
19	gregetan	geregetan	9	Benar
20	bhwa	bahwa	5	Benar
		berhawa	7	Salah
		bahariwan	9	Salah
21	kgn	kangen	6	Benar
		kognat	6	Salah
		keagaan	7	Salah
22	shbt	sahabat	7	Benar
		sahibulbait	11	Salah
		sahibulhajat	12	Salah
23	nmpk	nampak	6	Benar
		namplok	7	Salah
		nama produk	10	Salah
24	pd	pada	4	Benar
		padi	4	Salah
		padu	4	Salah
25	cmn	cuman	5	Benar
		campin	6	Salah
		cemani	6	Salah
26	krywn	karyawan	8	Benar
		karyawan lepas	13	Salah
		karyawan tetap	13	Salah
27	mlm	malam	5	Benar
		malim	5	Salah
		maklum	6	Salah
28	kmudian	kemudian	8	Benar
		kemudi haluan	12	Salah
		kemudian hari	12	Salah
29	bkn	beken	5	Salah
		bikin	5	Salah
		bukan	5	Benar
30	belajr	belajar	7	Benar
		berlajur	8	Salah
		berpelajaran	12	Salah
31	thun	tahun	5	Benar
		tahunan	7	Salah
		terhuni	7	Salah

32	lbih	lebih	5	Benar
		lebih jauh	9	Salah
		lebih lagi	9	Salah
33	trnyata	ternyata	8	Benar
		ternyatakan	11	Salah
34	adlh	adalah	6	Benar
		adihulung	9	Salah
		alhamdulillah	13	Salah
35	tmpt	tempat	6	Benar
		tumpat	6	Salah
		trompet	7	Salah
36	mmbuat	membuat	7	Benar
		membulat	8	Salah
		membulati	9	Salah
37	skolah	sekolah	7	Benar
		sekolahan	9	Salah
		sekolah raja	11	Salah
38	mnghasilkn	menghasilkan	12	Benar
39	trhdp	terhadap	8	Benar
		tata hidup	9	Salah
		tanah hidup	10	Salah
40	stlh	setelah	7	Benar
		salat loha	9	Salah
		sementelah	10	Salah
41	kpd	kepada	6	Benar
		kepadaan	8	Salah
		kepaduan	8	Salah
42	hti	hati	4	Benar
		hatif	5	Salah
		henti	5	Salah
43	slmt	selamat	7	Benar
		selimut	7	Salah
		selomot	7	Salah
44	mnjdi	menjadi	7	Benar
		main judi	8	Salah
		menjidari	9	Salah
45	dlm	dalam	5	Benar
		dalem	5	Salah
		dilam	5	Salah
46	lhir	lahir	5	Benar
		lanhir	6	Salah
		lahirjah	8	Salah
47	jd	jadi	4	Benar
		jeda	4	Salah
		judi	4	Salah
48	aplg	apalagi	7	Benar
		apologi	7	Salah
		apa lagi	8	Salah
49	mmg	mamang	6	Salah

		memang	6	Benar
		momong	6	Salah
50	bnci	banci	5	Salah
		benci	5	Benar
		bancik	6	Salah
51	tp	tap	3	Salah
		tip	3	Salah
		top	3	Salah
		tapa	4	Salah
		tapi	4	Benar
52	karna	karena	6	Benar
		karina	6	Salah
		karmina	7	Salah
53	seorg	seorang	7	Benar
		sejoreng	8	Salah
		sendorong	10	Salah
54	dr	dar	3	Salah
		dor	3	Salah
		dur	3	Salah
		dara	4	Salah
		dari	4	Benar
55	wjh	wajah	5	Benar
		wijdaniah	9	Salah
		wajah buku	10	Salah
56	tnp	tanpa	5	Benar
		tancap	6	Salah
		tanggap	6	Salah
57	mcm	macam	5	Benar
		mencema	7	Salah
		mencium	7	Salah
58	dpt	dapat	5	Benar
		depot	5	Salah
		dempet	6	Salah
59	mlht	melihat	7	Benar
		masalahat	8	Salah
		melihati	8	Salah
60	pdukung	pendukung	9	Benar
		pendukungan	11	Salah
61	pltkus	politikus	9	Benar
		pelir tikus	11	Salah
62	lg	log	3	Salah
		laga	4	Salah
		lagi	4	Benar
		lago	4	Salah
		lagu	4	Salah
63	trus	terus	5	Benar
		tirus	5	Salah
		torus	5	Salah
64	syg	sayang	6	Benar

		sayung	6	Salah
		syagar	6	Salah
65	jgn	jogan	5	Salah
		jagang	6	Salah
		jagoan	6	Salah
		jagung	6	Salah
		jangan	6	Benar
66	smpai	sampai	6	Benar
		simpai	6	Salah
		simpati	7	Salah
67	mnerima	menerima	8	Benar
		menerima	11	Salah
68	yg	yang	4	Benar
		yoga	4	Salah
		yogi	4	Salah
		yogia	5	Salah
69	trjd	terjadi	7	Benar
		tersujud	8	Salah
		terwujud	8	Salah
70	smua	semua	5	Benar
		semuka	6	Salah
		semula	6	Salah
71	knpa	kenapa	6	Benar
		kunarpa	7	Salah
		kenapang	8	Salah
72	hnya	hanya	5	Benar
		hinayana	8	Salah
		hantu raya	10	Salah
73	akn	akan	4	Benar
		akun	4	Salah
		akang	5	Salah
74	sepnjg	sepanjang	9	Benar
		seperujung	11	Salah
		seperunjungan	13	Salah
75	psal	pasal	5	Benar
		psalm	5	Salah
		parsial	7	Salah
76	jntung	jantung	7	Benar
		juntrung	8	Salah
		jantungan	9	Salah
77	pkerja	pekerja	7	Benar
		pekerjaan	9	Salah
		punya kerja	11	Salah
78	mlah	malah	5	Benar
		meluah	6	Salah
		mullah	6	Salah
79	hsl	hasil	5	Benar
		hastel	5	Salah
		hospital	8	Salah

80	mndpt	mendapat	8	Benar
		mendapati	9	Salah
		mana dapat	10	Salah
81	dn	dan	3	Benar
		den	3	Salah
		din	3	Salah
		dana	4	Salah
82	negri	negari	6	Salah
		negeri	6	Benar
		negrito	7	Salah
83	sj	saja	4	Benar
		saji	4	Salah
		soja	4	Salah
		suji	4	Salah
		sajak	5	Salah
84	slagi	selagi	6	Benar
		selangit	8	Salah
		stalagmit	9	Salah
85	brapa	berapa	6	Benar
		berupa	6	Salah
		berapat	7	Salah
86	slalu	selalu	6	Benar
		silalatu	8	Salah
87	kluarga	keluarga	8	Benar
		kekeluargaan	12	Salah
88	tgggu	tunggu	6	Benar
		tangguh	7	Salah
		tangguk	7	Salah
89	dkt	dekat	5	Benar
		dekut	5	Salah
		dikit	5	Salah
90	tny	tanya	5	Benar
		tadinya	7	Salah
		trenyuh	7	Salah
91	psr	pasar	5	Benar
		paser	5	Salah
		pasir	5	Salah
92	hrng	harga	5	Benar
		haring	6	Salah
		hering	6	Salah
93	brita	barita	6	Benar
		berita	6	Salah
		berpita	7	Salah
94	kt	kit	3	Salah
		kait	4	Salah
		kata	4	Benar
95	rngka	rangka	6	Benar
		rengka	6	Salah
		rangkai	7	Salah

96	km	kam	3	Salah
		kim	3	Salah
		kom	3	Salah
		kama	4	Salah
		kami	4	Benar
97	drpd	daripada	10	Benar
		dokter pribadi	14	Salah
98	cntk	cantik	6	Benar
		cantek	6	Salah
		cantrik	7	Salah
99	rhsia	rahsia	6	Salah
		rahasia	7	Benar
100	mnrut	menurut	7	Benar
		mencerut	8	Salah
		mengerut	8	Salah

Lampiran 5

Data Uji 1

<i>Tweet</i> sebelum dinormalisasi	<i>Tweet</i> setelah dinormalisasi
b aja sih kataku aku tdk terlalu menyukai cuma mo banyakkin notif abis itu keluar tdk berperilaku tanggung jawab hihi	b aja sih kataku aku tidak terlalu menyukai cuma (mo mok mol mop) banyak sedikitnya (nonaktif normatif) abis itu keluar tidak berperilaku tanggung jawab (hati-hati hila-hila)
sudhlah semua bru akan di mulai senin belajr jalani tanpa hrus berasumsi	sudhlah semua (baru beru biru bruk buru) akan di mulai senin belajar jalani tanpa hrus berasumsi
di saat politik kita begitu gaduh tak sedikit orang yg tetap bekerja dalam diam dan sepi dari tepuk tangan kades pon	di saat politik kita begitu gaduh tak sedikit orang yang tetap bekerja dalam diam dan sepi dari tepuk tangan kepala desa pekan olahraga nasional
akhirnya dia mnikah juga slmt pengantin baru	akhirnya dia menikah juga (selamat selimut selomot) pengantin baru
4 thun lbih indonesia cuman mnghasilkn pencitraan yg mroket janji ekonomi yg mroket trnyata oh kasihan	4 tahun lebih indonesia cuman menghasilkan pencitraan yang meroket janji ekonomi yang meroket ternyata oh kasihan
ilmu ibarat air sekolah pula adlh tmpt mmbuat timbapelajar yg mnamatkn sekolah brjaya mnghasilkn timbatimba itu	ilmu ibarat air sekolah pula adalah (tempat tumpat) membuat timbapelajar yang menamatkan sekolah berjaya menghasilkan timbatimba itu
jika pemerintah dan polri terlalu toleran thdp pembuat amppenyebarkan hoax ujaran kebencian	jika pemerintah dan polisi republik indonesia terlalu toleran terhadap pembuat amppenyebarkan hoax ujaran kebencian
mohon kpd pihak cyber polri untuk menindak akun ini karena tweet 2 ya bertentangan dengan sila pertama	mohon kepada pihak cyber polisi republik indonesia untuk menindak akun ini karena tweet 2 ya bertentangan dengan sila pertama
jadi stlh membaca ilustrasi tadi kita pasti jadi sadar bhw apapun yang kita mkn dan dibawa oleh darah jadi krusial apapun	jadi setelah membaca ilustrasi tadi kita pasti jadi sadar bahwa apapun yang kita (makan makin makna mukun) dan dibawa oleh darah jadi krusial apapun

balas rasa perdulimu kpd orang yg memperdulikanmu sebelum berusaha memperdulikan orang yang bahkan tidak peduli tentangmu kagome	balas rasa perdulimu kepada orang yang memperdulikanmu sebelum berusaha memperdulikan orang yang bahkan tidak peduli tentangmu kategori primer
indonesia ini damai banget kalo bikin salah yg menyinggung pihak lain tinggal minta maaf atau bilang saya cuma bercan	indonesia ini damai banget kalo bikin salah yang menyinggung pihak lain tinggal minta maaf atau bilang saya cuma (bercanda bercaran bercoang)
gengsi pada dasarnya adalah bentuk pertahanan ego ketidakmampuan menyesuaikan diri dengan keadaan yg seharusnya	gengsi pada dasarnya adalah bentuk pertahanan ego ketidakmampuan menyesuaikan diri dengan keadaan yang seharusnya
jadi yg diperhatiin kukunya bukan ga ada gandengannya	jadi yang diperhatiin kukunya bukan nggak ada gandengannya
am selamat siang kakak sekalian ada yg kenal saya sengaja ditutupin logo company biar pada ga tau g	am selamat siang kakak sekalian ada yang kenal saya sengaja ditutupin logo company biar pada nggak tahu g
kalo kaya gini tuh pasti yg disalahin cewe dibilang cewenya yg mau lah sama sama suka lah padahal yang bajinga	kalo kaya gini (tuah tuhu) pasti yang disalahin cewek dibilang cewenya yang mau lah sama sama suka lah padahal yang bajingan
gerombolan fpi hti gunakan pertemuan antara ormas dan pemerintah utk mengklaim bahwa bendera tauhid tdk dilarang pemerint	gerombolan (fonotipi fotokopi) hati gunakan pertemuan antara organisasi massa dan pemerintah untuk mengklaim bahwa bendera tauhid tidak dilarang pemerintah
mkn ayam kunyit mat rock tambah telur pehhh puasss hati den	(makan makin makna mukun) ayam kunyit mat rock tambah telur pehhh puasss hati den
ya mkn rumput ntr dibayar pke batu	ya (makan makin makna mukun) rumput (natar natur netra) dibayar pike batu
artinya bukan cm andalkan baca berita aja mbak coba sama 2 dicermati perda 2 yg ada luma	artinya bukan (cam cim) andalkan baca berita aja mbak coba sama 2 dicermati peraturan daerah 2 yang ada (lumai lumar lumas lumat)



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Kampus USU, Medan-20155
Telepon/Fax : 061-8228048 Laman: <http://fasilkom-ti.usu.ac.id>

No. : 660 /UN5.2.1.14/LLS/2019
Hal : Kesediaan Menguji

22 April 2019

Yth. Sdr.....
Tenaga Pendidik Fasilkom-TI USU
Medan

Dengan hormat, bersama ini diharapkan kesediaan Saudara untuk menguji Tugas Akhir (Ujian Komprehensif) dari mahasiswa:


Nama : Zurwatus Saniyah
NIM : 141402005
Program Studi : S1 Teknologi Informasi

Yang akan dilaksanakan pada :

Hari / Tanggal : Kamis / 25 April 2019
Pukul : 09.00 WIB s/d selesai
Tempat : Ruang Seminar S-1 Program Studi Teknologi Informasi
Fasilkom-TI USU

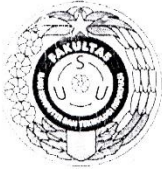
Demikian disampaikan, atas kehadiran Saudara tepat pada waktunya diucapkan terima kasih.

Wakil Dekan I


Dr. Elviawaty Muisa Zamzami, ST,MT,MM
NIP. 19700716 200501 2 002

Tembusan:

1. Dekan Fasilkom-TI USU
2. Ketua Program Studi S-1 Teknologi Informasi Fasilkom-TI USU
3. Bendahara Pembantu Pengeluaran Fasilkom-TI USU
4. Arsip



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
Jalan Universitas No. 9A Kampus USU, Medan-20155
Telepon/Fax : 061-8228048 Laman: <http://fasilkom-ti.usu.ac.id>

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
NOMOR: 666 /UN5.2.1.14/SK/LLS/2019

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi USU

- Membaca** : Surat permohonan mahasiswa Fasilkom-TI USU tanggal 7 Januari 2019 perihal permohonan ujian skripsi:
Nama : Zurwatus Saniyah
NIM : 141402005
Program Studi : S-1 Teknologi Informasi
Judul Skripsi : Normalisasi Mikroteks Berbentuk Singkatan Pada Teks Twitter Berbahasa Indonesia Menggunakan Algoritma Longest Common Subsequences
- Memperhatikan** : Bahwa mahasiswa tersebut telah memenuhi kewajiban dan persyaratan untuk ikut dalam pelaksanaan ujian skripsi mahasiswa pada Program Studi S-1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara T.A. 2017/2018.
- Menimbang** : Bahwa permohonan tersebut di atas dapat disetujui dan perlu ditetapkan dengan surat keputusan.
- Mengingat** : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
3. Keputusan Rektor USU Nomor 701/UN5.1.R/SK/SPB/2013 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 832/UN5.1.R/SK/SDM/2016 Tanggal 16 Mei 2016 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2016-2021.

MEMUTUSKAN

- Menetapkan Pertama** : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:
- | | |
|-----------------|---|
| Ketua | : Romi Fadillah Rahmat, B.Comp.Sc, M.Sc
NIP. 19860303 201012 1 004 |
| Sekretaris | : Dr. Sawaluddin, M.IT
NIP. 19591231 199802 1 001 |
| Anggota Penguji | : Dani Gunawan, ST, MT
NIP. 19820915 201212 1 002 |
| Anggota Penguji | : Ainul Hizriadi, S.Kom, M.Sc
NIP. 19851027 201706 1 001 |
| Moderator | : - |
| Panitera | : - |
- Kedua** : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2018.
- Ketiga** : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

Ditetapkan di Medan
Pada Tanggal: 22 April 2019
Dekan,



Opim Salim Sitompul
NIP. 19610817 198701 1 001

- Tembusan:
1. Ketua Program Studi S-1 Teknologi Informasi
 2. Yang bersangkutan
 3. Arsip