



**PENERAPAN *TEXT MINING* UNTUK VISUALISASI
FLUKTUASI HARGA KOMODITAS PANGAN (STUDI
KASUS TWITTER, LIPUTAN 6, DETIKCOM)**

TUGAS AKHIR

Program Studi

S1 Sistem Informasi

Oleh:

RAKHA BEAVIS LUCKYANO

15410100011

FAKULTAS TEKNOLOGI DAN INFORMATIKA

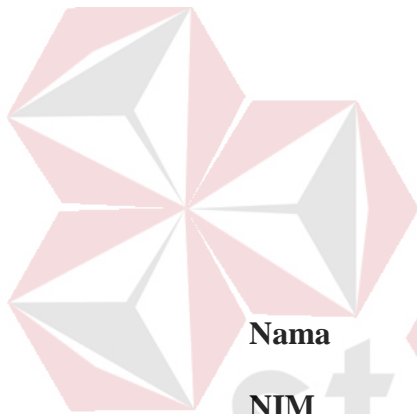
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2019

**PENERAPAN *TEXT MINING* UNTUK VISUALISASI FLUKTUASI
HARGA KOMODITAS PANGAN (STUDI KASUS TWITTER, LIPUTAN
6, DETIKCOM)**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer**



Oleh :

Nama : Rakha Beavis Luckyano
NIM : 15410100011
Program : S1 (Strata Satu)
Jurusan : Sistem Informasi

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

2019

TUGAS AKHIR
PENERAPAN *TEXT MINING* UNTUK VISUALISASI FLUKTUASI
HARGA KOMODITAS PANGAN (STUDI KASUS TWITTER, LIPUTAN
6, DETIKCOM)

dipersiapkan dan disusun oleh

Rakha Beavis Luckyano

NIM : 15.41010.0011

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada : Juli 2019

Susunan Dewan Penguji

Pembimbing

I. **Valentinus Roby Hananto, S.Kom., M.Sc.,**

NIDN. 0715028903


II. **Vivine Nurcahyawati, M.Kom.,**

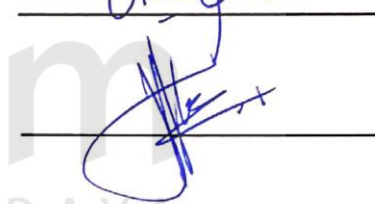
NIDN. 0723018101

Penguji

I. **Tutut Wuriyanto, M.Kom.**

NIDN. 0703056702


15/08/19



Tugas Akhir ini telah diterima sebagai salah satu persyaratan



untuk memperoleh gelar Sarjana
FAKULTAS TEKNOLOGI
DAN INFORMATIKA

stikom
SURABAYA

Dr. Jusak

Dekan Fakultas Teknologi dan Informatika

15/8/19

FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA



Scanned with
CamScanner

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Rakha Beavis Luckyano
NIM : 15410100011
Program Studi : S1 Sistem Informasi
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **PENERAPAN *TEXT MINING* UNTUK VISUALISASI
FLUKTUASI HARGA KOMODITAS PANGAN (
STUDI KASUS TWITTER, LIPUTAN 6, DETIKCOM)**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 11 Juli 2019

Yang menyatakan



Rakha Beavis Luckyano

Nim : 15410100011



Scanned with
CamScanner

“Nothing is Impossible”



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

"For My Dearest Mbuk"



ABSTRAK

Gagalnya produksi komoditas pangan dapat menyebabkan fluktuasi dan meroketnya harga dari komoditas pangan. Meski sudah tersedia alternatif media sosial dan media berita online untuk mengawasi fluktuasi harga tersebut, tetapi data yang ada tidak terstruktur, serta harus dibaca dan dipahami terlebih dahulu.

Penelitian ini bertujuan untuk membangun aplikasi berbasis *desktop* yang menerapkan teknik *text mining* dan algoritma Jaro-Winkler kepada Twitter, Liputan 6, dan Detikcom untuk memperoleh dan memvisualisasikan data harga komoditas pangan. penulis juga akan mengambil data pembandingan dari situs Pusat Informasi Harga Pangan Strategis Nasional (PIHPS).

Berdasarkan hasil dari penelitian, aplikasi berhasil memvisualisasikan fluktuasi harga komoditas pangan. Algoritma murni hanya mencari nama komoditas dan harga dari komoditas, dan tidak membandingkan faktor lain seperti satuan berat. Dari hasil uji akurasi terlihat bahwa tingkat akurasi rata-rata aplikasi adalah 47%. Sumber dengan tingkat akurasi tertinggi adalah sumber detik dengan tingkat akurasi 56%.

Kata Kunci: *Textmining, Python, Jaro-Winkler, Komoditas Pangan, Visualisasi*

KATA PENGATAR

Segala puja dan puji syukur kehadiran Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya lah penulis dapat menyelesaikan laporan proyek tugas akhir jurusan sistem informasi berjudul “Penerapan *Text Mining* Untuk Visualisasi Fluktuasi Harga Komoditas Pangan (Studi Kasus Twitter, Liputan 6, Detikcom)” dengan lancar. Penyelesaian laporan tugas akhir ini merupakan salah satu prasyarat untuk menempuh kelulusan.

Penulis mengucapkan rasa terimakasih yang sebanyak banyaknya kepada pihak-pihak dibawah ini yang telah membimbing, membantu dan mendukung penulis hingga terselesaikannya penulisan laporan ini.

1. Orang tua dan keluarga yang selama ini selalu memberi doa dan dukungan.
2. Bapak Valentinus Roby Hananto, S.Kom., M.Sc., selaku Pembimbing I yang selalu meluangkan waktu dan dengan sabar membimbing serta membantu dalam pembuatan laporan dan aplikasi pada proyek tugas akhir ini
3. Ibu Vivine Nurcahyawati, M.Kom., selaku Pembimbing II yang telah meluangkan waktu dan tidak bosan-bosan menghadapi pertanyaan-pertanyaan penulis serta membimbing dan membantu dalam penyusunan aplikasi dan laporan proyek tugas akhir ini.
4. Diar Pasahari, Dewana Wira, Daffa Akbar, serta teman teman dari grup Private, yang telah mendukung dan memberi semangat kepada penulis selama ini.
5. Seluruh rekan-rekan mahasiswa stikom angkatan 2015 yang telah memberi dukungan kepada penulis.

6. Pihak-pihak lain yang tidak bisa disebutkan satu-persatu yang telah memberikan dukungan secara moral ataupun material.

Penulis menyadari bahwa masih banyak sekali kekurangan didalam penyusunan laporan tugas akhir ini. Untuk itulah penulis sangat terbuka terhadap saran maupun kritik yang membangun. Semoga laporan ini dapat memberi manfaat kepada pembaca dan individu-individu yang berkepentingan.

Surabaya, Juli 2019

Penulis



DAFTAR ISI

ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2. Perumusan Masalah	4
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian	5
1.5. Manfaat	5
BAB II LANDASAN TEORI	7
2.1 Penelitian Terdahulu	7
2.2. <i>Python</i>	7
2.3. Microsoft SQL Server	9
2.4. Text Mining.....	11
2.4.1. Information Retrieval (IR)	12
2.4.2 <i>Data Mining</i>	14
2.4.3. Natural Language Preprocessing (NLP)	16
2.5. Visualisasi	18
2.6. Sumber <i>Text Mining</i>	19
2.6.1. Twitter	19
2.6.2. Detik.com	19
2.6.3 Liputan6.com	20
2.7 Waterfall System Development Life Cycle	20
BAB III METODE PENELITIAN.....	23
3.1 Communication.....	23
3.1.1 Studi Literatur	23
3.1.2. Observasi.....	24
3.2. Planning.....	24
3.3. Modeling	24

3.3.1 Analisis Kebutuhan	25
3.3.2 Pemodelan	28
3.3.3 Desain <i>User Interface</i> (UI).....	46
3.4. Construction	49
3.4.1 Penerapan Text Mining	49
3.4.2 Testing.....	59
3.5. Deployment	66
BAB IV HASIL DAN PEMBAHASAN	67
4.1 Implementasi	67
4.1.1 Implementasi Sistem	68
4.1.2 Implementasi <i>User Interface</i> (UI)	72
4.2 Uji Coba	76
4.2.1 Uji Coba Fungsi	76
4.2.2 Uji Akurasi	81
4.3 Evaluasi	86
4.3.1. Karakteristik Data <i>Source</i>	88
4.3.2 Karakteristik Data Komoditas.....	90
BAB V PENUTUP.....	94
5.1 Kesimpulan	94
5.2 Saran.....	95
DAFTAR PUSTAKA	96
LAMPIRAN	99

DAFTAR TABEL

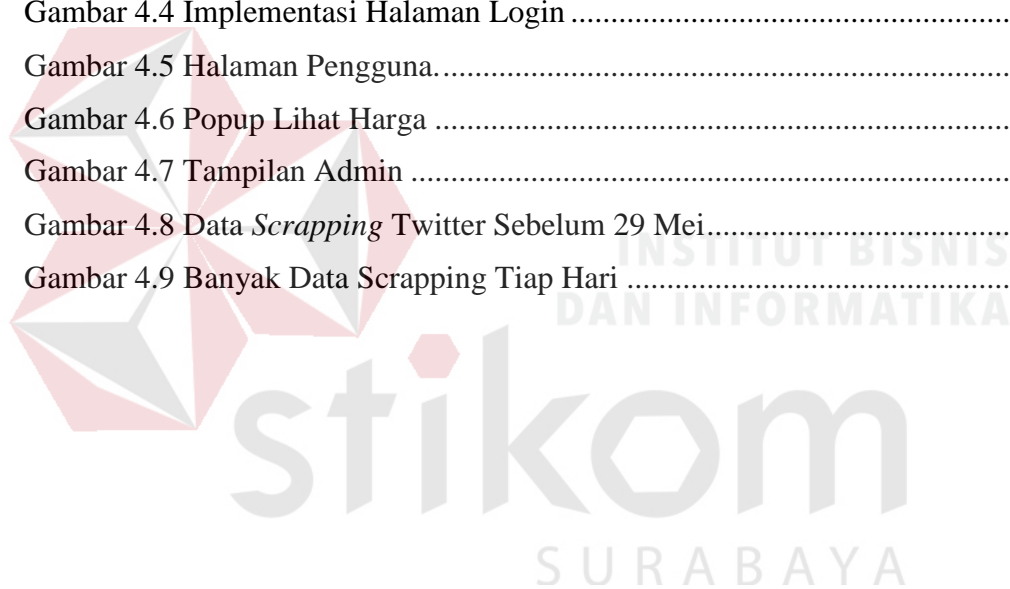
Tabel 3.1 Jadwal Pengerjaan.....	24
Tabel 3.2 Tabel Skenario Testing Fungsi	59
Tabel 3.3 Skenario Uji Akurasi Sumber Twitter	63
Tabel 3.4 Skenario Uji Akurasi Sumber Media Berita Online	64
Tabel 3.5 Skenario Akurasi Jaro Winkler	65
Tabel 4.2 Implementasi Testing Fungsi	76
Tabel 4.3 Uji Akurasi Sumber Twitter.....	82
Tabel 4.4 Uji Akurasi Sumber Media Berita Online.....	84
Tabel 4.5 Contoh Perhitungan Akurasi Rata-Rata.....	86
Tabel 4.6 Akurasi Jaro Winkler	86
Tabel 4.7 Data Yang Didapat Dari <i>Scrapping</i>	88
Tabel 4.8 Jumlah Data Harga Komoditas Yang Didapat	91
Tabel 4.9 Detail Penyebaran Data Perkomoditas Twitter	91
Tabel 4.10 Detail Penyebaran Data Perkomoditas Detik.....	92
Tabel 4.11 Detail Penyebaran Data Perkomoditas Liputan 6	93



DAFTAR GAMBAR

Gambar 2.1 Fungsi Pyjarowinkler.	18
Gambar 2.2 <i>Waterfall System Development Life Cycle</i>	21
Gambar 3.1 SDLC Waterfall.....	23
Gambar 3.2 Use case diagram.....	26
Gambar 3.3 Activity Diagram Login	28
Gambar 3.4 Activity Diagram Scrapping Web.....	29
Gambar 3.5 Activity Diagram <i>Scrapping</i> Twitter	30
Gambar 3.6 Activity Diagram <i>Web Processing</i>	31
Gambar 3.7 Activity Diagram <i>Tweet Processing</i>	32
Gambar 3.8 Activity Diagram <i>Import Data Pembanding</i>	33
Gambar 3.9 Activity Diagram Lihat Harga	33
Gambar 3.10 Sequence Diagram <i>Login</i>	34
Gambar 3.11 Sequence Diagram <i>Scrapping</i> Web.....	35
Gambar 3.12 Sequence Diagram <i>Scrapping</i> Twitter	37
Gambar 3.13 Sequence Diagram <i>Web Processing</i>	39
Gambar 3.14 Sequence Diagram <i>Tweet Processing</i>	41
Gambar 3.15 Sequence Diagram <i>Import Data Pembanding</i>	43
Gambar 3.16 Sequence Diagram Lihat Harga	44
Gambar 3.17 <i>Class Diagram</i>	46
Gambar 3.18 Desain Halaman <i>Login</i>	47
Gambar 3.19 Desain Halaman Admin	47
Gambar 3.20 Desain Halaman Pengguna.....	48
Gambar 3.21 Desain Halaman <i>Popup</i> Lihat Harga.....	48
Gambar 3.22 Kode Pembersihan Tag Web.....	49
Gambar 3.23 Kode Pembersihan Tag Twitter.....	50
Gambar 3.24 Kode Casefolding	50
Gambar 3.25 Kode Stemming	51
Gambar 3.26 Kode <i>Filtering</i>	51
Gambar 3.27 Kode <i>Tokenize</i> Kalimat	52
Gambar 3.28 Kode <i>Tokenize</i> Kata	52
Gambar 3.29 List Komoditas	53

Gambar 3.30 Kode Pemecahan Kata Komoditas	53
Gambar 3.31 Awal Proses Seleksi Pada Web	54
Gambar 3.32 Kode Penomoran Kata	54
Gambar 3.33 Kode Pencarian Kata Pertama Nama Komoditas.....	55
Gambar 3.34 Pencarian Sisa Kata Nama Komoditas Didalam Berita.	56
Gambar 3.35 Kode Penggunaan Database “cek_tweet”	56
Gambar 3.36 Kode Pencarian kata Rp.	57
Gambar 3.37 Kode penyimpanan harga komoditas.	57
Gambar 4.1 Visualisasi Harga Daging Ayam Ras	68
Gambar 4.2 Visualisasi Harga Daging Sapi.....	69
Gambar 4.3 Visualisasi Harga Telur Ayam Ras	70
Gambar 4.4 Implementasi Halaman Login	72
Gambar 4.5 Halaman Pengguna.....	73
Gambar 4.6 Popup Lihat Harga	74
Gambar 4.7 Tampilan Admin	75
Gambar 4.8 Data <i>Scrapping</i> Twitter Sebelum 29 Mei.....	89
Gambar 4.9 Banyak Data Scrapping Tiap Hari	90



DAFTAR LAMPIRAN

Lampiran 1. Syntax Scrapping Twitter	99
Lampiran 2. Hasil Scrapping Twitter.....	101
Lampiran 3. Syntax Preprocessing Twitter	101
Lampiran 4. Hasil Preprocessing Twitter.....	103
Lampiran 5. Syntax Seleksi Jaro-Winkler Twitter.....	104
Lampiran 6. Hasil Seleksi Jaro-Winkler Twitter	107
Lampiran 7. Syntax Scrapping Web (Detik).....	107
Lampiran 8. Syntax Preprocessing Web (Detik).....	109
Lampiran 9. Syntax Seleksi Jaro-Winkler Web (Detik)	109



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada tahun 2019 ini sudah tidak dapat dipungkiri lagi bahwa efek *global warming* telah mengakibatkan perubahan-perubahan ekstrim, baik dari segi iklim dan suhu yang kita semua dapat rasakan. *Intergovernmental Panel on Climate Change* (IPCC) memprediksi peningkatan temperatur global rata rata global akan meningkat 1,1 hingga 6,4 °C pada 1990 dan 2100, dampak dari pemanasan global (*global warming*) akan mempengaruhi pola presipitasi, evaporasi, *water run-off*, kelembapan tanah dan variasi iklim yang sangat fluktuatif secara keseluruhan dapat mengancam keberhasilan produksi pangan (Ismail, 2018).

Gagalnya produksi komoditas pangan dapat menyebabkan fluktuasi dan meroketnya harga dari komoditas pangan itu sendiri. Selain itu menurut Gubernur BI Perry Warjiyo yang dikutip CNN Indonesia (2018) kenaikan harga komoditas pangan akan mempengaruhi inflasi perekonomian di Indonesia. Maka dari itu kemampuan pemerintah untuk mengawasi fluktuasi perkembangan harga komoditas yang berfluktuatif sangatlah penting. Karena dari data harga tersebut pemerintah dapat mendapatkan masukan untuk merumuskan kebijakan yang perekonomian yang tepat demi menghindari inflasi perekonomian Indonesia.

Salah satu media yang dapat digunakan untuk mengawasi fluktuasi harga komoditas pangan adalah internet. Internet bukan lagi barang yang mewah bagi masyarakat Indonesia. Saat ini populasi masyarakat Indonesia telah mencapai sekitar 262 juta orang dan menurut Asosiasi Penyelenggara Jasa Internet

Indonesia(APJII) yang dikutip oleh Kompas(2018) lebih dari 50 persen atau kurang lebih 143 juta orang sudah dapat menikmati akses internet pada tahun 2017.

Kemajuan dibidang internet ini juga dimanfaatkan oleh lembaga lembaga resmi pemerintah untuk menerbitkan situs web yang dapat meningkatkan kinerja mereka. Salah contohnya adalah situs milik Badan Pusat Statistik (BPS). Menurut situs resmi BPS (2019) BPS adalah lembaga resmi pemerintah yang bertanggung jawab langsung kepada presiden dan salah satu fungsi dari BPS adalah menyediakan kebutuhan data bagi pemerintah dan masyarakat. Sayangnya fungsi ini kurang berjalan dengan baik. Dari Observasi yang dilakukan penulis , situs BPS lebih fokus terhadap penerbitan data statistik dalam bentuk dokumen, sehingga tidak ada diagram dalam bentuk untuk memvisualisasikan perubahan harga komoditas secara langsung. Selain itu BPS hanya menerbitkan data dalam dua jenis kurun waktu yaitu pertahun atau perbulan, dan itupun masih ada kemungkinan bahwa data statistik tersebut batal dipublikasikan. Hal tersebut juga akan berdampak terhadap kecepatan dan ketepatan dari pengambilan keputusan pemerintah Indonesia sendiri

Mudahnya akses internet inilah yang mengakibatkan tingginya pertumbuhan media sosial dan juga perpindahan media berita konvensional menjadi media berita online. Salah satu media sosial yang seringkali digunakan masyarakat untuk mengutarakan pendapat mereka adalah Twitter. Pinegar(2018) menjelaskan Twitter adalah *platform* media sosial dan berita online yang terdiri dari kumpulan profil dan *newsfeed*. Menurut Dwi Ardiansyah, *Head of Business Development Twitter* untuk *South-East Asia & Australia* yang dikutip dalam Viva (2018) twitter di Indonesia sering digunakan sebagai sarana interaksi bagi

masyarakat dan instansi komersil dan instansi pemerintahan. Selain itu twitter juga dianggap sebagai informasi secara cepat di Indonesia. Sedangkan dari sisi media berita online, menurut survey yang dilakukan Alexa yang dikutip dalam Aseanup (2017) Liputan 6 dan Detik merupakan situs media berita online dengan jumlah *traffic* tertinggi di Indonesia.

Meskipun sudah tersedia alternatif media sosial dan media berita online untuk mengawasi fluktuasi harga komoditas pangan, tetap saja data yang tersedia masih tidak terstruktur dan berbentuk artikel panjang yang masih harus dibaca dan dipahami terlebih dahulu. Hal ini akan membuat pembaca membuang waktu yang cukup lama untuk dapat menemukan informasi yang relevan. Perbandingan harga pun akan sulit dilakukan karena data yang hanya berbentuk teks.

Berdasarkan data dan permasalahan diatas, agar BPS mengetahui harga *real* yang ada ditemui masyarakat dipasaran dan dapat membandingkannya dengan data pemerintah, serta demi membantu kelengkapan dan kesuksesan perilisan dokumen statistik komoditas pangan oleh BPS, penulis memutuskan untuk membangun aplikasi berbasis *desktop* yang akan menerapkan teknik *text mining* dan algoritma Jaro-Winkler kepada Twitter, Liputan 6, dan Detik.com untuk memperoleh dan memvisualisasikan data harga komoditas pangan. Selain itu penulis juga akan mengambil data pembanding dari situs Pusat Informasi Harga Pangan Strategis Nasional (PIHPS) sebagai data pembanding. Adapun komoditas pangan yang akan divisualisasikan adalah 3 komoditas pangan yang memiliki tingkat konsumsi tertinggi pada tahun 2016 per-kategori yang dicantumkan didalam buku statistik pertanian 2017 yang diterbitkan oleh Kementrian Pertanian.

1.2. Perumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang dapat diangkat dalam penelitian ini adalah bagaimana menerapkan *text mining* untuk visualisasi fluktuasi harga komoditas pangan?

1.3. Batasan Masalah

Untuk menghindari kesalahpahaman ruang lingkup dari persoalan yang akan dijabarkan dalam perancangan aplikasi ini, penulis membatasi permasalahan yang dibahas berdasarkan perumusan masalah diatas menjadi beberapa poin berikut ini :

1. Aplikasi yang dibangun berbentuk *desktop*
2. Data sekunder diperoleh dari media sosial (Twitter) dan media berita online(Liputan 6 dan Detik.com)
3. Sumber data pembanding diambil dari situs Pusat Informasi Harga Pangan Strategis Nasional.
4. Objek penelitian adalah 3 komoditas dengan tingkat konsumsi tertinggi per-kategori pada tahun 2016 yang disebutkan didalam buku Statistik Pertanian (2017) .Yaitu; bayam, daging ayam ras, daging ayam kampung, daging sapi, jagung, jeruk, kangkung, ketela rambat, rambutan,semangka, ketela pohon, talas,telur ayam ras,telur ayam kampung,telur puyuh,tepung terigu,tomat.
5. Data simulasi diambil mulai dari tanggal 8 April 2019 - 23 Juni 2019
6. Data harga yang diambil adalah yang berbentuk numeric dan memiliki nominal “Rp” Contoh :Rp 50.000
7. Tingkat akurasi jaro-winkler yang diterapkan adalah 0,90 atau 90%.

8. Aplikasi murni mencari nama komoditas dan harganya tanpa mempertimbangkan faktor lain, seperti satuan harga dan apakah komoditas itu impor atau tidak.

1.4. Tujuan Penelitian

Berdasarkan latar belakang dan rumusan permasalahan diatas, maka tujuan dari penelitian ini adalah untuk menerapkan *text mining* untuk visualisasi fluktuasi harga komoditas pangan.

1.5. Manfaat

Adapun manfaat yang diharapkan dapat didapatkan dari penelitian ini adalah, BPS diharapkan dapat membandingkan harga nyata yang dijumpai masyarakat dipasaran dengan data yang mereka dapat melalui survey yang dilakukan. Selain itu diharapkan penelitian ini dapat menjadi gerbang untuk meningkatkan automasi penerapan *text mining*, kecerdasan buatan, serta aplikasi-aplikasi pintar lain yang lebih mutakhir dimasa yang akan datang

1.6. Sistematika Penulisan

Adapun sistematika penulisan yang diterapkan penulis dalam penelitian ini adalah

BAB I PENDAHULUAN

Pada bab ini penulis membahas latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini penulis memaparkan teori-teori yang akan diterapkan selama pengerjaan penelitian ini.

BAB III METODE PENELITIAN

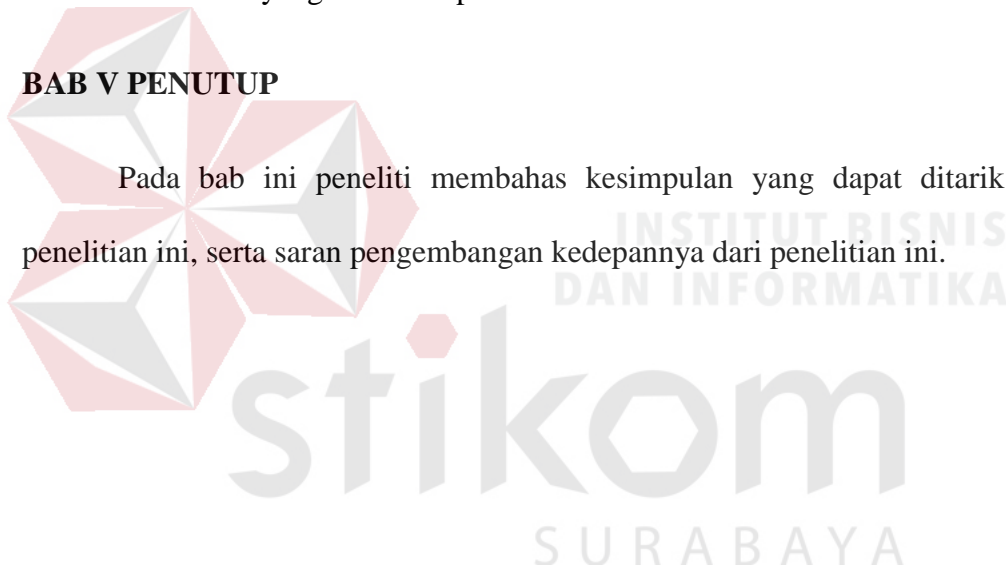
Pada bab ini peneliti menjelaskan langkah-langkah yang dilakukan dalam penelitian ini. Adapun langkah-langkah tersebut dibagi menjadi 5 langkah besar yaitu; *Communication, Planning, Modeling, Construction*, dan *Deployment*.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini peneliti memaparkan hasil dari implementasi langkah-langkah yang telah dilakukan pada bab sebelumnya. Selain itu peneliti juga akan melakukan evaluasi dari hasil yang telah didapatkan.

BAB V PENUTUP

Pada bab ini peneliti membahas kesimpulan yang dapat ditarik dari penelitian ini, serta saran pengembangan kedepannya dari penelitian ini.



BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Adapun beberapa penelitian terdahulu yang digunakan sebagai referensi dalam penyusunan penelitian ini antara lain adalah ; penelitian (Saifullo, 2017) yang menerapkan *natural language processing* untuk menjawab secara otomatis pertanyaan dari calon pelanggan, penelitian (Wardani, 2019) yang menerapkan *web scrapping* dan metode *naïve bayes* untuk menganalisis sentimen dan melakukan pemeringkatan situs belanja online di Indonesia, Penelitian (Kressa, 2019) yang membangun aplikasi yang menggunakan *web scrapping* dan algoritma *jaro-winkler* untuk mencari beasiswa di internet.

2.2. Python

Python adalah bahasa pemrograman *open source* yang bermulti guna dan bermulti paradigma dengan dukungan untuk struktur pemrograman fungsional, prosedural dan fungsional. Python biasa dipakai untuk program *standalone* dan untuk *scripting* aplikasi di berbagai macam *platform* dan dianggap sebagai salah satu bahasa pemrograman yang paling banyak dipakai didunia (Lutz, 2014).

Fitur-fitur didalam bahasa pemrograman *Python* menekankan pada keterbacaan kode dan fungsionalitas library dan desain bahasa pemrograman yang mengoptimalkan produktivitas pengembang aplikasi, kualitas *software*, portabilitas program, dan integrasi dari komponen program. Program *Python* dapat dijalankan di berbagai platform yang biasa digunakan, termasuk Unix, Linux, Windows, Macintosh, Java, .NET, Android, iOS dan masih banyak lagi

Dibawah ini merupakan kelebihan dan keterbatasan bahasa pemrograman *Python* menurut Mindfire Solutions (2017) :

Berikut merupakan kelebihan bahasa pemrograman *Python*:

1. *Library* yang Bervariasi

Python menyediakan library pemrograman besar yang mencakup area seperti operasi *string*, internet, web service tools , serta antarmuka dan protokol untuk sistem operasi. Banyak dari tugas pemrograman telah dituliskan didalam *script* yang ada didalam *library* sehingga meminimalisir banyak kode yang harus ditulis di *Python*

2. Fitur Integrasi

Python mengintegrasikan Enterprise Application Integration yang mempermudah pengembangan web service dengan memanggil komponen COM atau COBRA. *Python* juga dapat memproses XML dan bahasa markup lainnya.

3. Meningkatkan Produktifitas *Programmer*

Besarnya library programan *Python* dan desain yang mendukung stuktur programming berorientasi objek meningkatkan produktivitas *programmer* dibandingkan saat menggunakan bahasa seperti Java, VB, Perl, C, C++ dan C#

Adapun keterbatasan bahasa pemrograman *Python* antara lain :

1. Kesulitan menggunakan Bahasa Pemrograman Lain

Pecinta bahasa pemrograman *Python* tebiasa dengan fitur dan *library* yang disediakan, sehingga mereka kesusahan mempelajari atau bekerja menggunakan bahasa pemrograman lain

2. Lemah dalam Komputasi Mobile

Python sudah terbiasa digunakan pada platform desktop dan server. Tetapi *Python* dianggap sebagai bahasa yang lemah untuk komputasi mobile sehingga jarang sekali aplikasi *mobile* dibangun menggunakan *Python*.

3. Kecepatan yang Lambat

Python di eksekusi bukan melalui *compiler* tetapi menggunakan bantuan *interpreter*, hal ini memperlambat proses eksekusi *Python*. Tetapi *Python* masih cepat untuk implementasi aplikasi web

4. Run-time Errors

Bahasa pemrograman *python* ditulis secara dinamis sehingga dilaporkan banyak error yang muncul sehingga membutuhkan waktu testing yang lebih lama

2.3. Microsoft SQL Server

Menurut Djuandi (2002) SQL Server adalah sebuah sistem arsitektur terbuka yang memungkinkan para pengembang program memperluas dan menambahkan fungsi-fungsi ke dalam database tersebut. Sedangkan Andri Kunio Kusrini (2007) berpendapat bahwa SQL Server adalah perangkat lunak *relational database management system* (RDBMS) yang didesain untuk melakukan proses manipulasi database berukuran besar dengan berbagai fasilitas.

Berikut merupakan kelebihan dan kekurangan dari penggunaan Microsoft SQL server menurut Moufarrege (2015):

Adapun kelemahan dari SQL Server adalah :

1. Biaya

Salah satu kelemahan terbesar dari penggunaan Microsoft SQL server dari *relational database management system* (RDBMS) lain adalah biaya lisensi yang cukup mahal. Meskipun penggunaan software untuk pengembangan atau tujuan pembelajaran tidak dipungut biaya, segala bentuk dari pemakaian untuk kebutuhan bisnis membutuhkan biaya untuk lisensi. Sebagai contohnya untuk SQL Server 2008 Standard Edition dipatok harga \$7,171 per prosessor.

2. Kompatibilitas yang Terbatas

Microsoft SQL server didesain untuk hanya dapat dieksekusi di *server* yang berbasis Windows. Untuk beberapa faktor termasuk biaya lisensi dan masalah keamanan, para pengembang biasa mengalihkan situs mereka ke *server* berbasis Unix yang mengakibatkan mereka tidak bisa menggunakan SQL server.

Berikut ini adalah kelebihan dari SQL Server :

3. Software Manajemen Tingkat Enterprise

Microsoft SQL Server menawarkan software manajemen database tingkat *professional dan enterprise* yang mudah digunakan dan memiliki banyak fitur. Software yang ditawarkan oleh Microsoft juga terintegrasi dengan kerangka kerja .NET

4. Dukungan Data *Recovery* yang baik

SQL Server memiliki banyak opsi untuk mencegah kehilangan data melalui fitur *log files*, *caching* dan *backup*.

2.4. Text Mining

Menurut Rouse (2018) *Text mining* adalah proses untuk menelusuri dan menganalisa data teks yang tidak terstruktur dalam jumlah yang besar, dengan bantuan perangkat lunak yang dapat mengidentifikasi konsep, pola, topik, kata kunci, dan atribut lainnya didalam data tersebut. *Text mining* dapat dikatakan hampir sama dengan *data mining*, tetapi lebih focus terhadap teks dari pada jenis data lain yang lebih terstruktur dengan memanfaatkan teknologi *natural language processing* (NLP) yang menerapkan prinsip linguistik komputasional dalam menguraikan dan menafsirkan rangkaian data. Sedangkan Wijaya (2018) berpendapat bahwa *text mining* merupakan teknologi *artificial intelligence* (AI) yang memungkinkan penggunanya untuk mengubah konten inti dari sebuah dokumen teks menjadi data kuantitatif secara cepat. Adapun area penerapan text mining menurut Dataflair (2018) antara lain:

1. *Information Retrieval* (IR)

Sistem IR berguna untuk memperoleh rangkaian dokumen yang berkaitan dengan permasalahan yang dialami pengguna dengan mengaplikasi algoritma *text mining* yang kompleks terhadap sekumpulan dokumen.

2. *Data Mining*

Data mining dapat dideskripsikan sebagai proses untuk mencari pola yang tersembunyi didalam data.

3. *Natural Language Preprocessing* (NLP)

NLP adalah istilah yang dikaitkan dengan kegiatan mendesain metode dan algoritma pemrosesan yang akan memproses bahasa manusia secara komputasional.

4. *Information Extraction*(IE)

IE adalah kegiatan untuk mengekstrak informasi terstruktur dari sumber data teks / dokumen yang tidak terstruktur maupun semi terstruktur. Proses ini biasa dilakukan dengan metode NLP.

2.4.1. Information Retrieval (IR)

Berikut merupakan proses Information Retrieval yang akan dilakukan dalam penelitian ini :

A. Web Scrapping

Menurut Syahab (2017) *web scrapping* adalah proses ekstraksi sebuah dokumen semi terstruktur dari internet yang umumnya berupa halaman-halaman web dalam bahasa markup seperti html atau xhtml dan menganalisisnya untuk mendapatkan data tertentu dari halaman tersebut. *Software* atau *script* web *scrapping* dapat mengakses *World Wide Web* (www) secara langsung dengan menggunakan *Hypertext Transfer Protocol* atau melalui *browser web*. Proses *Scrapping* dapat dilakukan secara manual oleh user atau secara otomatis menggunakan *bot* (*web crawler*). Didalam penelitian ini ada 2 modul yang akan digunakan untuk *scrapping* yaitu :

A.1. Beautiful Soup

Menurut Leonard Richardson (2015) Beautiful Soup merupakan salah satu *library* yang ada didalam bahasa pemrograman *python*. *Beautiful Soup* mengambil

data dari html dan xml. Hal itu menyebabkan *Beautiful Soup* sering dipakai dalam proses *web scrapping* dalam bahasa pemrograman *Python*. Menurut dokumentasi yang ada di situs resmi beautiful soup Crummy (2015) menjelaskan bahwa beautiful soup merubah objek dokumen html kompleks menjadi pohon objek Python yang kompleks. Tetapi adapun objek dan metode yang akan ditemui adalah 4 yaitu :

a. Tag

Tag adalah objek yang berkorespondensi dengan html atau xml yang ada di dokumen asli. Fitur terpenting dari sebuah tag adalah nama yang dimiliki tag tersebut, dan atributnya.

b. Navigable String

Navigable String adalah metode yang digunakan *beautiful soup* untuk mengambil sebuah string yang berkorespondensi kepada teks yang ada didalam tag.

c. BeautifulSoup

Objek *beautiful soup* merepresentasikan dokumen secara keseluruhan. Objek ini dapat diperlakukan sebagai tag objek dalam hampir seluruh skenario. Objek ini tidak memiliki nama ataupun atribut.

d. Comment

Objek *comment* merupakan sebuah sub kelas dari *navigable string*.

Selain metode dan fungsi diatas adapun fungsi yang juga akan digunakan dalam proses pengambilan data dari web antara lain adalah fungsi *find_all* yang akan digunakan untuk mencari semua informasi / konten didalam tag yang ditentukan

A.2. Tweepy

Tweepy menurut Jason Ridgen (2018) adalah sebuah *library* untuk mengakses API twitter yang cocok digunakan untuk automasi dan membuat *bot* twitter. Menurut situs dokumentasi resmi Tweepy (2009) sebelum dapat menggunakan Tweepy perlu dilakukan autentikasi yang menggunakan fungsi OAuthHandler yang akan memasukkan *consumer token* dan *secret* yang didapat dari Twitter.

Setelah melakukan autentikasi pengguna dapat memakai berbagai kelas API yang ada didalam Tweepy yang memberi akses kepada seluruh metode RESTful API yang ada didalam Twitter. Salah satu API yang akan digunakan dalam pencarian *tweet* antara lain adalah fungsi *api.search* yang akan mencari *tweet* dari seluruh user dalam kurun waktu 1 minggu

Setiap metode dapat menerima berbagai parameter dan memberi respon dari parameter tersebut. Saat metode API dijalankan, sering kali respon yang diberikan kepada pengguna adalah objek permodelan kelas dari Tweepy. Objek ini berisi data yang dikirim dari Twitter yang akan dapat digunakan didalam aplikasi yang akan dibuat ataupun dijalankan.

2.4.2 Data Mining

Berikut merupakan proses data mining yang akan diterapkan dalam penelitian ini :

A. Text Preprocessing

Text preprocessing adalah salah satu aktifitas yang dilakukan setelah ekstraksi data yang bertujuan untuk memastikan data lebih terstruktur. Berikut

merupakan beberapa langkah *text preprocessing* yang dijelaskan dalam Informatikalogi (2016) :

1. *Case Folding*

Case Folding digunakan untuk mengubah keseluruhan teks dalam suatu sumber menjadi huruf kecil (*lowercase*) agar proses pencarian informasi tidak terhambat ketidak konsistenan dalam penggunaan huruf kapital. Sebagai contoh kumpulan kata berikut; “CABAI”, “Cabai|”, “caBai”, dan “CabAi”, akan menghasilkan hasil retrieval yang sama, yaitu “cabai”.

2. *Tokenizing*

Tokenizing adalah tahap pemotongan data string yang dimasukkan berdasarkan kata yang menyusunnya. Sebagai contoh kalimat “aku suka makan ikan dan sayuran” akan dipotong menjadi sekumpulan kata; “aku”, ”suka”, “makan”, ”ikan”, “dan”, “sayuran”.

3. *Filtering*

Filtering adalah tahap mengambil kata penting dari hasil proses *tokenizing*. Hal itu dapat dilakukan dengan menggunakan algoritma *stoplist*. Contoh dari *Stoplist/stopword* adalah “yang”, ”dan”, “di”, “dari”, dan lain sebagainya. Hasil *tokenizing* yang sudah disebutkan pada tahap sebelumnya akan difilter menjadi kumpulan kata ; “aku”, ”suka”, “makan”, ”ikan”, “sayuran”.

4. *Stemming*

Proses *stemming* dilakukan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen dan untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk yang berbeda karena mendapatkan imbuhan yang berbeda. Sebagai contoh kumpulan kata; “melukai”, “telukai”, dan ”dilukai” akan dirubah menjadi kata “luka”

Dalam proses text preprocessing ini akan diterapkan dengan menggunakan modul NLTK dan PySastrawi. Menurut situs dokumentasi resmi NLTK(), NLTK adalah *platform* yang digunakan untuk membangun aplikasi python yang menggunakan bahasa manusia. NLTK menyediakan desain antar muka yang mudah untuk digunakan dan memiliki lebih dari 50 sumber *corpora* dan *lexical* seperti Wordnet, dan peralatan lain untuk *text preprocessing* untuk klasifikasi, *tokenization*, *stemming*, dan lain sebagainya. Beberapa fungsi yang akan digunakan dalam modul ini antara lain adalah ; *tokenize* untuk tokenization, *corpus import stopwords* untuk filtering. Sedangkan PySastrawi digunakan untuk proses filtering dan stemming dalam bahasa Indonesia.

2.4.3. Natural Language Preprocessing (NLP)

Berikut merupakan proses NLP yang diterapkan dalam penelitian ini:

A. Algoritma Jaro-Winkler

Setelah melalui proses *text preprocessing*, data akan diseleksi kemiripannya dengan algoritma Jaro-Winkler untuk mendapat data yang relevan. Menurut Kurniawati, Sulistyono dan Sazali (2010) algoritma Jaro-Winkler distance

adalah salah satu variasi dari Jaro *distance metric* yang biasa digunakan untuk mengukur kesamaan dari dua string. Semakin tinggi Jaro-Winkler *distance* diantara dua string yang dibandingkan, maka semakin mirip kedua string tersebut. Nilai normalnya adalah 0 yang berarti tidak sama dan 1 yang menandakan kesamaan.

Algoritma Jaro-Winkler dibagi menjadi 3 bagian :

1. Menghitung panjang string.
2. Menentukan jumlah karakter yang sama didalam dua string.
3. Menentukan jumlah transposisi (t)

Berikut ini merupakan rumus yang digunakan untuk menghitung Jaro-Winkler *distance*. Rumus pertama digunakan untuk menghitung jarak (dj) antara dua string S1 dan S2 dengan memperhatikan jumlah karakter yang sama persis (m). Untuk lebih jelasnya dapat dilihat dibawah ini:

$$dj = \frac{1}{3} \left(\frac{m}{S1} + \frac{m}{S2} + \left(\frac{m-t}{m} \right) \right)$$

Setelah jarak (dj) antara 2 string ditemukan, langkah selanjutnya adalah menghitung Jaro-Winkler *distance* (dw). Hal ini dilakukan dengan memperhatikan panjang karakter yang sama sebelum ditemukan ketidaksetaraan (l) dan konstanta *scaling factor* (p) yang menurut Winkler adalah 0,1. Adapun rumus perhitungannya adalah :

$$dw = dj + (l \times p (1 - dj))$$

Didalam penerapan algoritma ini, penulis akan memakai modul yang bernama pyjarowinkler yang pada situs dokumentasinya di Pypi.org(2019) dikembangkan

menggunakan rumus yang sama seperti diatas. Adapun kode dari fungsi yang akan digunakan adalah sebagai berikut

```
from pyjarowinkler import distance
jw=distance.get_jaro_distance('string 1','String 2', winkler=True, scaling=0.1)
```

Gambar 2.1 Fungsi Pyjarowinkler.

Adapun string yang akan dibandingkan akan dimasukkan didalam tanda petik dan hasil dari fungsi tersebut adalah hasil perhitungan rumus algoritma yang sudah dijelaskan sebelumnya.

2.5. Visualisasi

Menurut Shneiderman (1998) visualisasi merupakan penggunaan komputer pendukung untuk menggambarkan data secara visual dan interaktif untuk memperkuat pengamatan. Susanto (2015) berpendapat bahwa visualisasi data adalah teknik menyajikan data secara visual melalui grafik, chart, atau peta agar terlihat menarik dan tetap informatif. Sedangkan Jubilee Enterprise(2015) menyebutkan bahwa visualisasi data adalah upaya untuk mengubah data-data teks menjadi bentuk visual yang lebih mudah untuk dimengerti. Maka dapat diambil kesimpulan bahwa visualisasi adalah sebuah upaya untuk menampilkan data secara visual melalui grafik, chart, atau peta dengan menggunakan komputer pendukung untuk memudahkan pemahaman dari data tersebut.

Untuk penelitian ini jenis grafik yang sering digunakan adalah grafik garis (*line*). Hal ini dikarenakan Grafik garis biasa dipergunakan untuk memvisualisasikan data yang berupa *trend* atau perkembangan suatu informasi yang diamati dari waktu ke waktu dan juga cocok untuk digunakan sebagai perbandingan nilai. Dengan menggunakan grafik garis fluktuasi dari trend objek

yang diteliti serta perbandingan yang dilakukan dalam suatu kurun waktu tertentu akan lebih terlihat dibandingkan grafik seperti grafik lingkaran.

2.6. Sumber *Text Mining*.

Berikut merupakan sumber media sosial dan situs media berita online yang akan menjadi sumber dari penerapan *text mining*:

2.6.1. Twitter

Twitter memulai debutnya pada bulan Maret 2006 sebagai proyek sampingan sebuah perusahaan *podcast* di San Francisco. Lima tahun kemudian, pada September 2011 Twitter mengumumkan bahwa mereka berhasil meraih 100 juta pengguna yang aktif setiap bulannya, 400 juta orang pengunjung *website* per bulannya dan melayani milyaran pesan per minggunya di seluruh dunia. Twitter merupakan salah satu saluran komunikasi penting dalam kejadian politik dan bencana alam, selain itu bisnis juga mengandalkan twitter sebagai sarana marketing, promosi dan pelayanan pelanggan. Menurut O'Reilly dan Milstein (2012) Twitter mempunyai kesamaan dengan media komunikasi online sudah ada, tetapi yang membuatnya unik adalah kemudahan untuk menulis dan membaca postingan yang ada dikarenakan panjang maksimal karakter yang hanya 280 kata.

2.6.2. Detik.com

Detik.com adalah sebuah situs berita dan artikel online. Saat ini detik.com hanya merilis berita secara online dan bergantung terhadap iklan yang ada didalam situs mereka untuk keuntungan. Detik.com pertama kali didirikan oleh Budi Darsono, Abdul Rahman dan Didi Nigrahadi pada tanggal 9 juli 1998. Salah satu faktor yang mendongkrak popularitas detik.com pada masa itu adalah

peliputan kerusakan Mei 1998 di Jakarta yang pada saat itu tidak ada media berita lain yang berani untuk meliputnya.

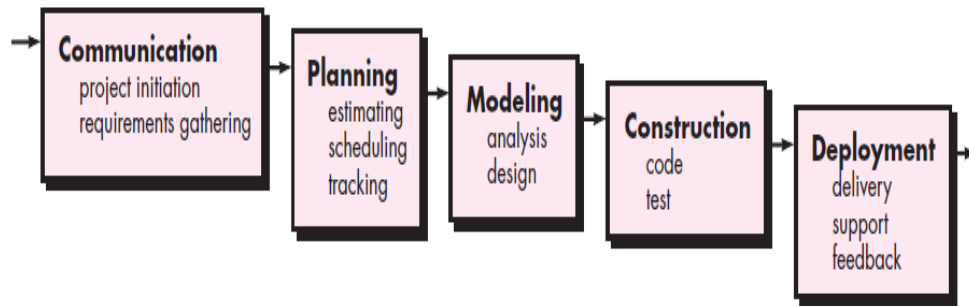
Pada tanggal 3 Agustus 2011 Detik.com diakuisisi oleh Chairul Tanjung, pemilik dari CT Corp, dengan total biaya 60 juta US dolar dari PT Agranet Multicitra Siberkom. Setelah itu, Detik.com beroperasi dibawah manajemen Trans Media dengan Suroyo Bimantoro sebagai komisaris utama.

2.6.3 Liputan6.com

Liputan6.com berdiri sejak 24 Agustus 2000. Awalnya Liputan6.com hanya menyajikan berita yang sudah tayang di stasiun televisi pada program Liputan6 SCTV(Surya Citra Televisi). Lalu pada tanggal 24 Mei 2012, perusahaan induk PT. Elang Mahkota Teknologi Tbk. ,yang merupakan perusahaan terbuka, memutuskan untuk memisah Liputan 6.com dari SCTV dengan membentuk perusahaan baru bernama PT. Kreatif Media Karya (KMK). Sejak itulah Liputan6.com berubah menjadi portal media online dengan beragam pilihan berita yang diberikan dan menjadi salah satu jajaran 5 portal berita online terbesar di Indonesia. (Liputan6, 2019)

2.7 Waterfall System Development Life Cycle

Menurut Pressman (2015) mengungkapkan bahwa metode *waterfall* adalah metode pengembangan sistem informasi terstruktur atau disebut dengan *classic life cycle*. Metode *waterfall* menurut Pressman (2015) memiliki 5 tahapan yaitu :



Gambar 2.2 *Waterfall System Development Life Cycle*

1). *Communication*

Pada tahap ini dilakukan penggalan informasi kepada pengguna. Dari langkah inilah akan ditemukan kebutuhan dari pengguna

2). *Planning*

Tahap planning meliputi perencanaan pengerjaan software yang meliputi pengerjaan teknis yang akan dilakukan, sumber yang dibutuhkan serta jadwal pengerjaan

3). *Modeling*

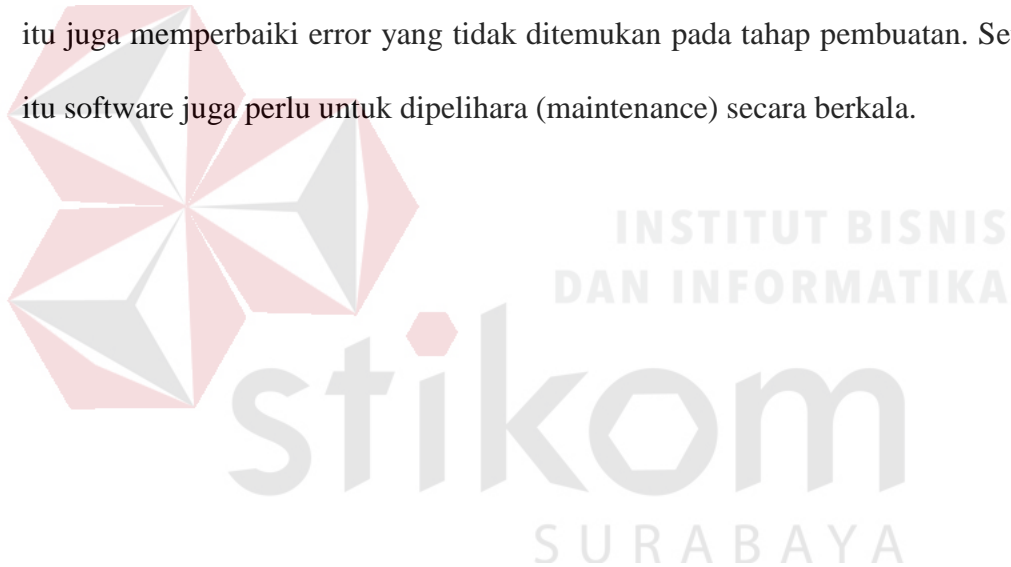
Tahap *modeling* berguna untuk mentransformasi perencanaan yang dilakukan pada proses *planning* menjadi rangkaian permodelan perencanaan perangkat lunak. Permodelan ini dapat terbagi menjadi rancangan struktur basis data, arsitektur dan diagram aliran perangkat lunak

4). Construction

Pada tahap ini perangkat lunak mulai dibangun melalui proses pembuatan kode(*code generation*). Setelah pembuatan kode selesai maka akan dilaksanakan *testing* terhadap aplikasi yang dibuat untuk memastikan perangkat lunak bekerja sesuai perencanaan yang telah dibuat.

5). Deployment

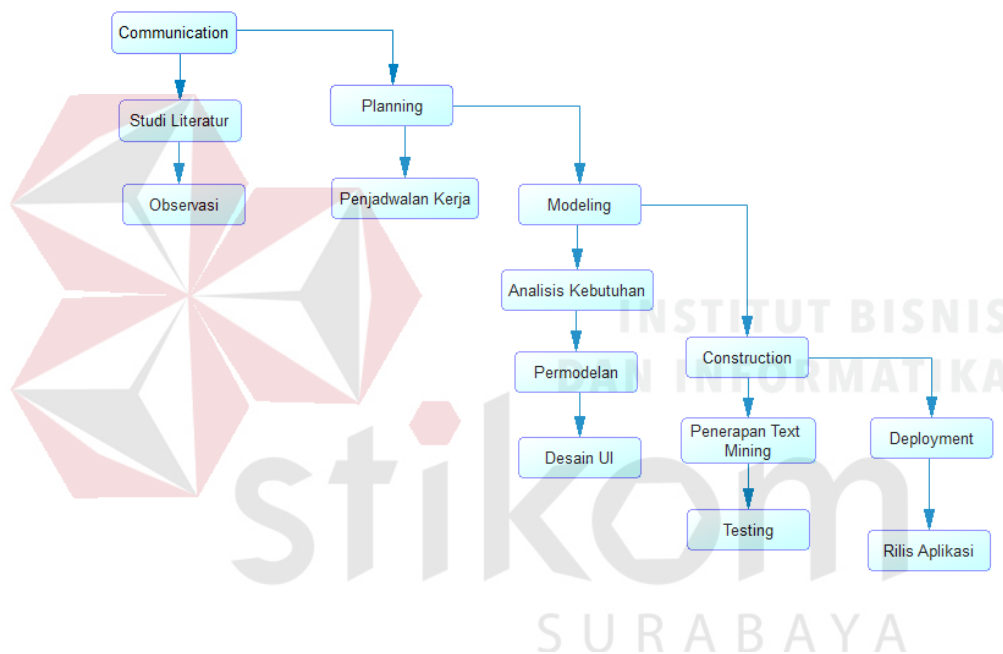
Dalam tahap deployment ini, sistem dipasang dan mulai digunakan. Selain itu juga memperbaiki error yang tidak ditemukan pada tahap pembuatan. Setelah itu software juga perlu untuk dipelihara (maintenance) secara berkala.



BAB III

METODE PENELITIAN

Dalam penelitian ini penulis memilih menggunakan *system development life cycle* Waterfall. Aplikasi akan dibangun dengan berorientasikan kepada objek sehingga desain sistem akan menggunakan UML. Berikut merupakan langkah langkah pengembangan dalam SDLC Waterfall:



Gambar 3.1 SDLC Waterfall

3.1 Communication.

Pada tahap ini penulis akan mengumpulkan informasi-informasi penting yang berkaitan dengan penelitian. Tahap ini terbagi menjadi dua yaitu :

3.1.1 Studi Literatur

Studi literatur dilakukan dengan tujuan untuk menggali pengetahuan lebih dalam terhadap teknik *text preprocessing*, algoritma jaro-winkler, pengembangan

aplikasi dan tentang sifat dan keadaan komoditas pangan di Indonesia. Studi literatur dilaksanakan dengan cara membaca jurnal ataupun dengan membaca situs web yang berkaitan dengan penelitian. Adapun sumber pembelajaran penulis akan dicantumkan didalam daftar pustaka.

3.1.2. Observasi

Dalam tahap ini penulis melakukan pengamatan terhadap informasi statistik komoditas pangan yang diberikan didalam situs BPS serta mengamati komoditas mana yang memiliki tingkat konsumsi terbesar pada tahun 2016 pada buku statistik pertanian 2017 BPS.

3.2. Planning

Pada tahap planning penulis melakukan penjadwalan pengerjaan penelitian baik dari sisi permodelan sistem dan *coding aplikasi*. Untuk lebih jelasnya penjadwalan akan di paparkan pada tabel dibawah ini.

Tabel 3.1 Jadwal Pengerjaan

No	Kegiatan	April				Mei				Juni				Juli
		1	2	3	4	1	2	3	4	1	2	3	4	1
1	<i>Communication</i>													
2	<i>Planning</i>													
3	<i>Modelling</i>													
4	<i>Construction</i>													
5	<i>Deployment</i>													

3.3. Modeling

Pada tahapan ini penulis melakukan penyusunan model aplikasi yang akan dibuat. Tahap *modeling* dibagi menjadi 3 tahap yaitu :

3.3.1 Analisis Kebutuhan

Pada tahap ini penulis menganalisis kebutuhan pengguna yang dibagi kembali menjadi beberapa tahap yaitu :

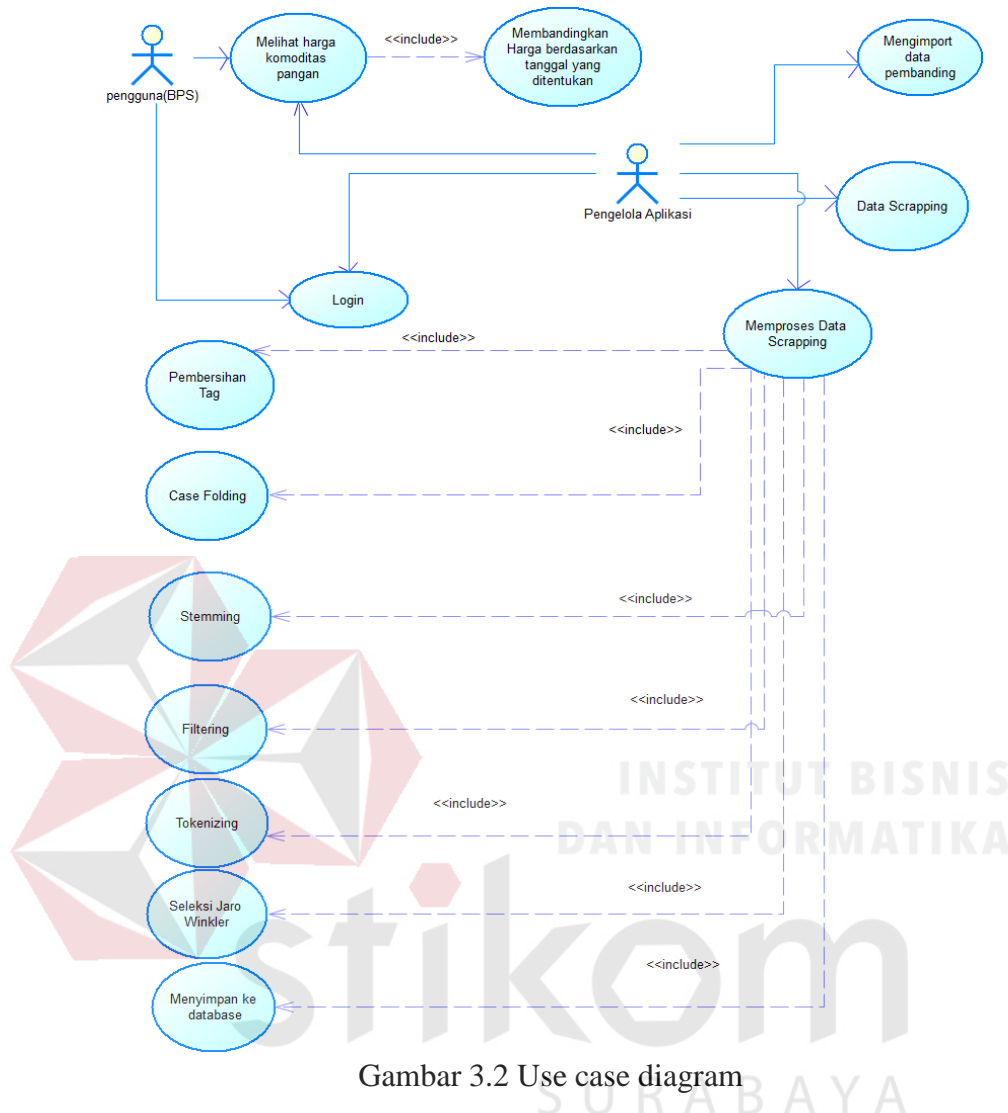
A. Identifikasi Permasalahan

Adapun permasalahan yang terjadi adalah bahwa semakin lama harga komoditas pangan akan semakin fluktuatif karena dipengaruhi oleh *global warming* dan menjadi penyebab inflasi perekonomian di Indonesia. Kemampuan pemerintah untuk dapat mengawasi fluktuasi harga komoditas akan berpengaruh kepada kualitas dan kecepatan perumusan kebijakan perekonomian untuk menanggulangi inflasi yang terjadi.

Peran Badan Pusat Statistik (BPS) sebagai penyedia data untuk pemerintah memegang peran penting dalam hal ini. Sayangnya situs web yang dimiliki BPS tidak memiliki grafik apapun yang dapat digunakan untuk mengawasi fluktuasi harga secara langsung, dan penerbitan dokumen statistik oleh bps memiliki interval waktu yang lama serta ada kemungkinan gagal terbit.

Meskipun pemerintah juga dapat mengawasi fluktuasi harga melalui media sosial dan media berita online, bentuk data yang ada tidak teratur dan dalam bentuk teks yang panjang sehingga membutuhkan waktu yang cukup lama pula untuk memahami teks tersebut.

B. Analisis Kebutuhan Pengguna



Gambar 3.2 Use case diagram

C. Analisis Kebutuhan Fungsional

Kebutuhan fungsional dari sistem dibagi menjadi 2 yaitu :

1. Fungsi *Text Mining*

Aplikasi dapat melakukan proses *text mining* yang terdiri dari ;

- 1) Ekstraksi data
- 2) Pembersihan tag
- 3) *Case Folding*
- 4) *Tokenizing*

5) *Filtering*

6) Seleksi Jaro-Winkler

7) Penyimpanan data ke *database*

8) Import data pembandingan

2. Fungsi Visualisasi Data

Aplikasi dapat memvisualisasikan data yang disimpan pada database. Data yang telah disimpan akan divisualisasikan menggunakan diagram garis pada aplikasi dengan sumbu x yang merupakan satuan waktu (tanggal, bulan) dan sumbu y untuk satuan harga dari komoditas. Dapat menampilkan beberapa data dari suatu parameter waktu. Jika ada beberapa harga dalam satu hari yang sama maka harga tersebut akan diambil rata-ratanya. Visualisasi data akan dibedakan berdasarkan sumber dari pengambilan data.

D. Analisis Kebutuhan Non-fungsional

Adapun kebutuhan non-fungsional dari sistem adalah :

1. *Security* (Keamanan)

Fitur login dan logout menggunakan *username* dan *password* bertujuan untuk memastikan pengguna yang ingin mengakses aplikasi benar benar pihak yang memiliki hak akses.

2. *Response Time*

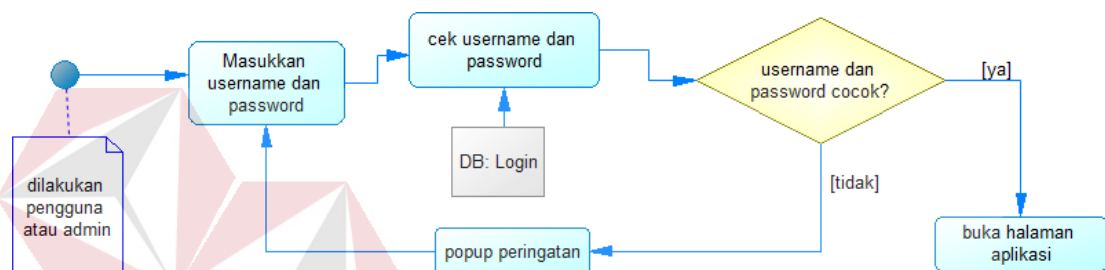
Waktu yang dilakukan untuk visualisasi data harga komoditas pangan tidak lebih dari 1 menit

3.3.2 Pemodelan

Pada tahap ini penulis akan menyusun pemodelan sistem dari aplikasi yang akan dibangun. Untuk fungsi scrapping data akan dibagi lagi menjadi dua permodelan yaitu scrapping web dan scrapping twitter karena mengalami proses yang sedikit berbeda. Adapun tahap pemodelan yang akan diterapkan antara lain :

A. Penyusunan *activity* diagram

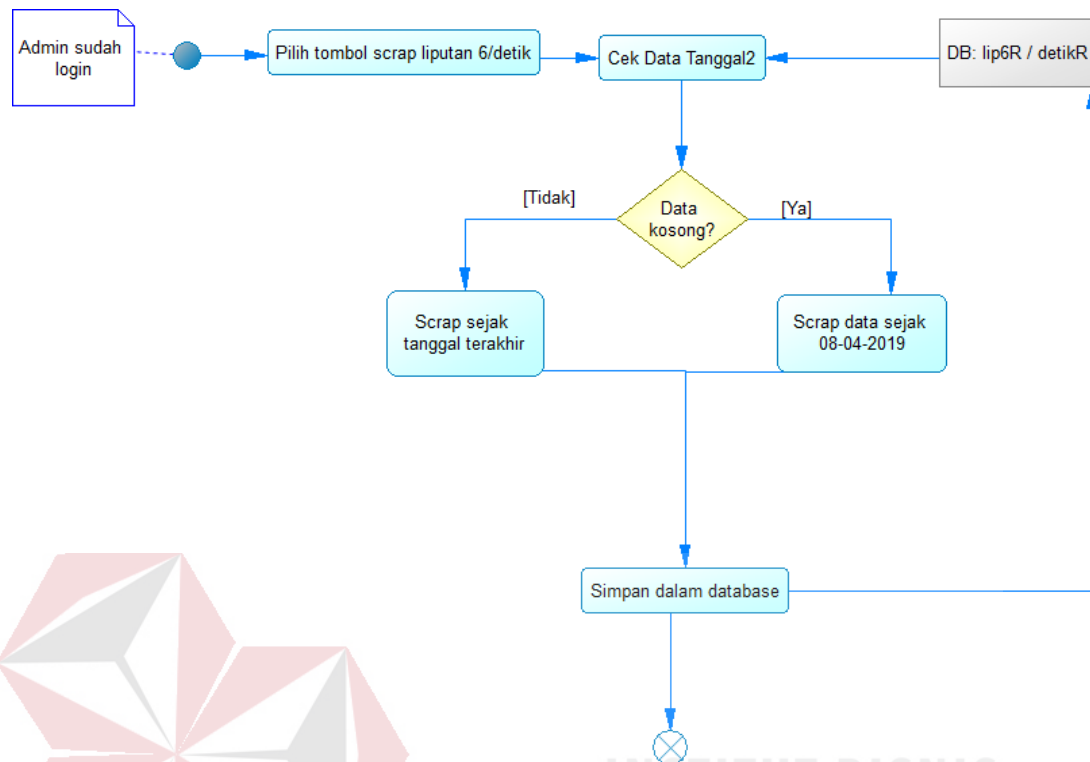
A.1. Activity Diagram Login



Gambar 3.3 Activity Diagram Login

Diagram diatas menggambarkan bahwa untuk melakukan proses login pengguna atau admin harus memasukkan username dan password untuk dapat mengakses aplikasi.

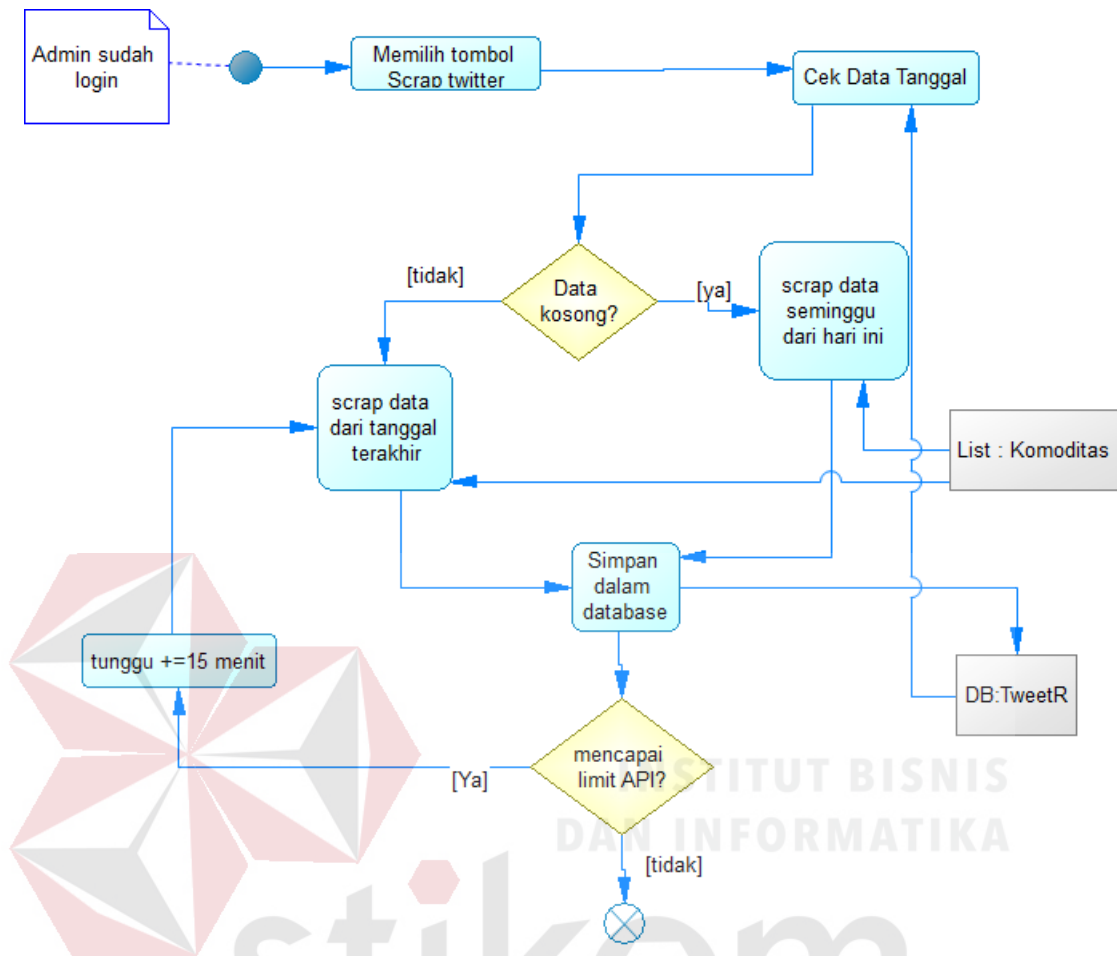
A.2. Activity Diagram Scrapping Web



Gambar 3.4 Activity Diagram Scrapping Web

Diagram diatas menunjukkan bahwa untuk melakukan *scrapping* situs web, admin perlu memilih tombol scrap liputan 6 / detik sesuai dengan sumber yang diinginkan. Jika tidak ada data di database maka *scrapping* akan dilakukan dari tanggal 8 april 2019, jika ada, aplikasi akan melakukan *scrapping* mulai dari tanggal terakhir.

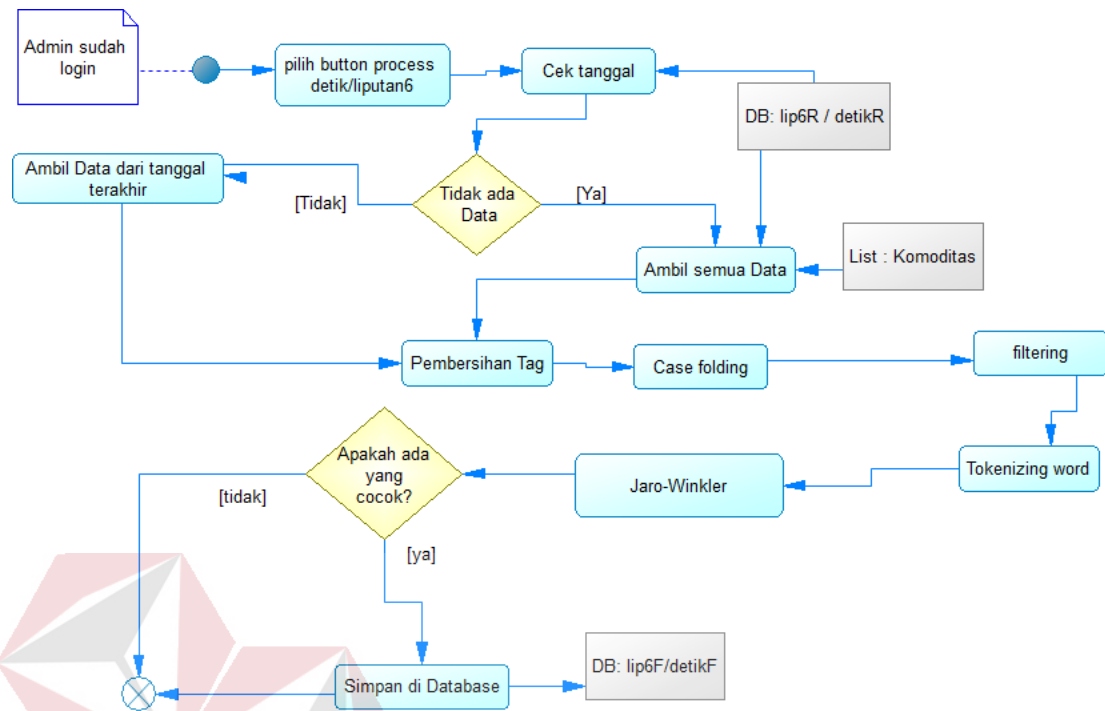
A.3. Activity Diagram Scrapping Twitter



Gambar 3.5 Activity Diagram Scrapping Twitter

Gambar diatas menggambarkan bahwa untuk melakukan *scrapping* Twitter, admin harus menekan tombol *Scrap* Twitter. Jika data *scrapping* terlalu banyak dan mencapai batas yang ditentukan oleh pihak Twitter, aplikasi akan memberi tahu pengguna untuk menunggu selama kurang lebih 15 menit sebelum melanjutkan proses *scrapping* secara otomatis.

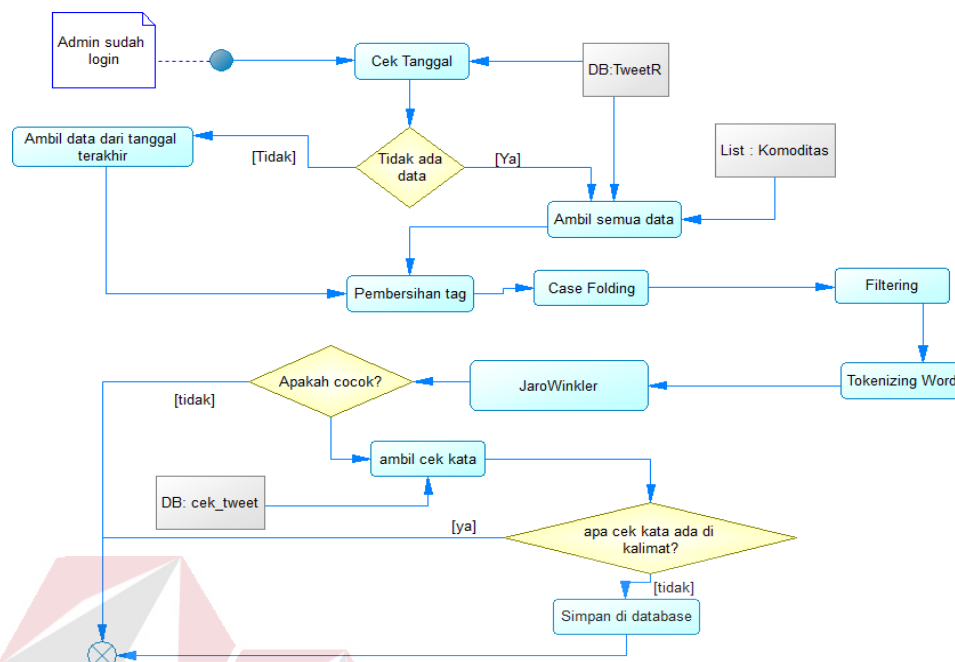
A.4. Activity Diagram Web Processing



Gambar 3.6 Activity Diagram Web Processing

Gambar diatas menjelaskan bahwa untuk memproses data mentahan hasil *scrapping* situs Liputan6 dan detik admin dapat memilih tombol *process* detik atau *process* liputan6 sesuai dengan sumber data yang diinginkan. Aplikasi akan melakukan pengecekan tanggal. Selanjutnya, aplikasi akan melakukan process *text preprocessing* dan seleksi jaro-winkler pada data. Jika data cocok, yang disini berarti mengandung nama komoditas dengan tingkat kemiripan 0,90 atau 90 persen dan memiliki data harga, maka data akan disimpan didalam *database*.

A.5. Activity Diagram *Tweet Processing*

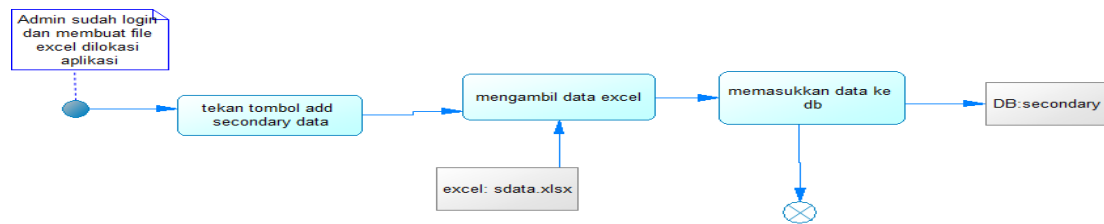


Gambar 3.7 Activity Diagram *Tweet Processing*

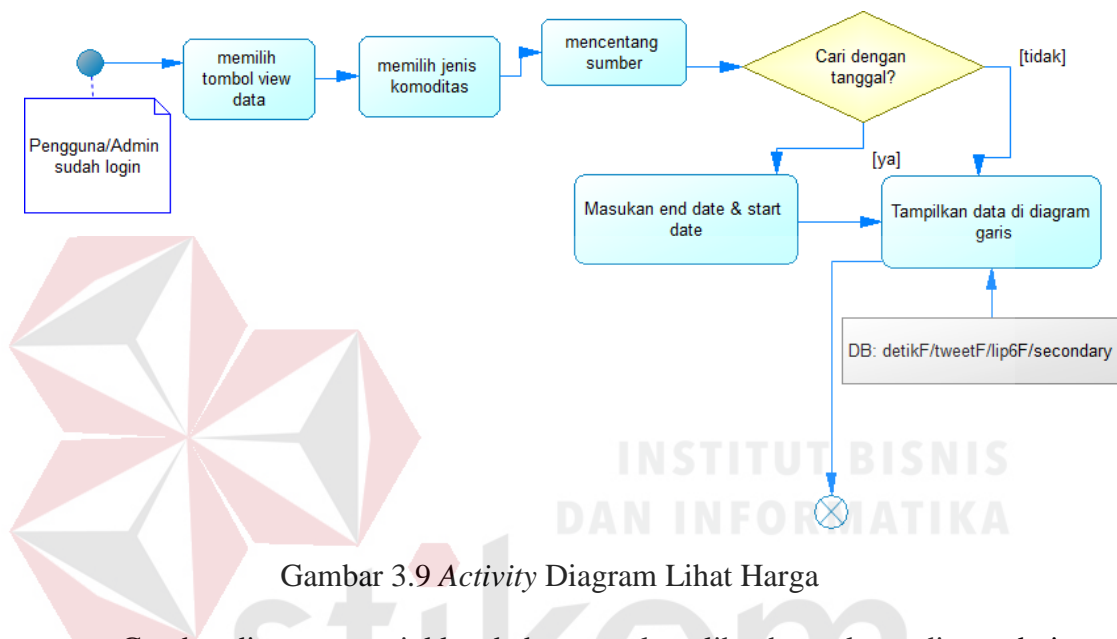
Gambar diatas menjelaskan bahwa untuk memproses data mentahan hasil *scrapping* Twitter admin dapat memilih tombol *process* twitter. Aplikasi akan melakukan pengecekan tanggal. Selanjutnya, aplikasi akan melakukan *process text preprocessing* dan seleksi jaro-winkler pada data. Jika data cocok, yang disini berarti mengandung nama komoditas dengan tingkat kemiripan 0,90 atau 90 persen dan memiliki data harga, maka data akan disimpan didalam *database*.

A.6. Activity Diagram *Import Data Pemandang*

Gambar 3.7 menjelaskan bahwa untuk meng-*import* data pemandangan, admin harus membuat file berbentuk excel yang disimpan dengan nama sdata pada direktori tempat aplikasi berada dengan kolom bernama tgl, harga, dan jenis. Setelah itu admin dapat login dan menekan tombol *add secondary data*.

Gambar 3.8 Activity Diagram *Import* Data Pembanding

A.7. Activity Diagram Lihat Harga

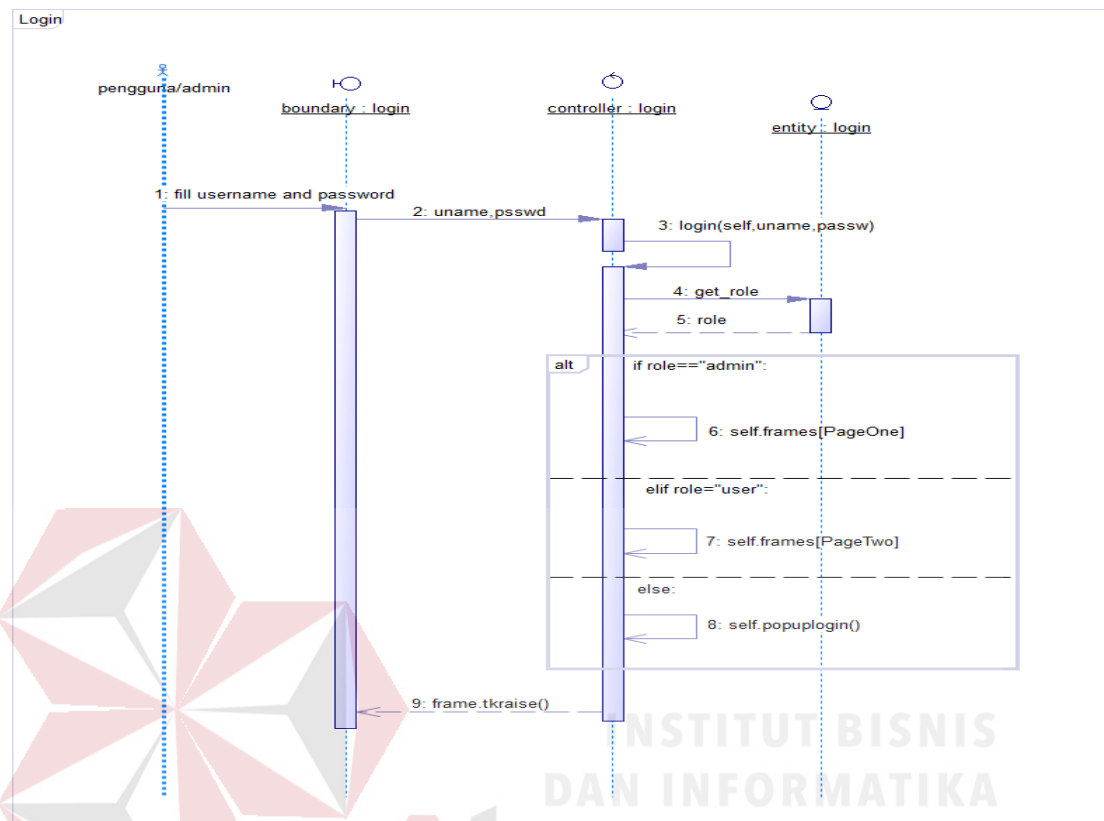


Gambar 3.9 Activity Diagram Lihat Harga

Gambar diatas menunjukkan bahwa untuk melihat harga komoditas, admin dan pengguna dapat memilih tombol lihat data. Setelah itu pengguna atau admin dapat memilih jenis dari komoditas yang akan dilihat dan mencentang sumber yang diinginkan termasuk data pembanding(secondary). Selain itu pengguna atau admin juga dapat memasukkan parameter opsional seperti mencari menggunakan tanggal ataupun menampilkan data pembanding. Data akan ditampilkan dalam bentuk diagram garis.

B. Penyusunan *sequence diagram*

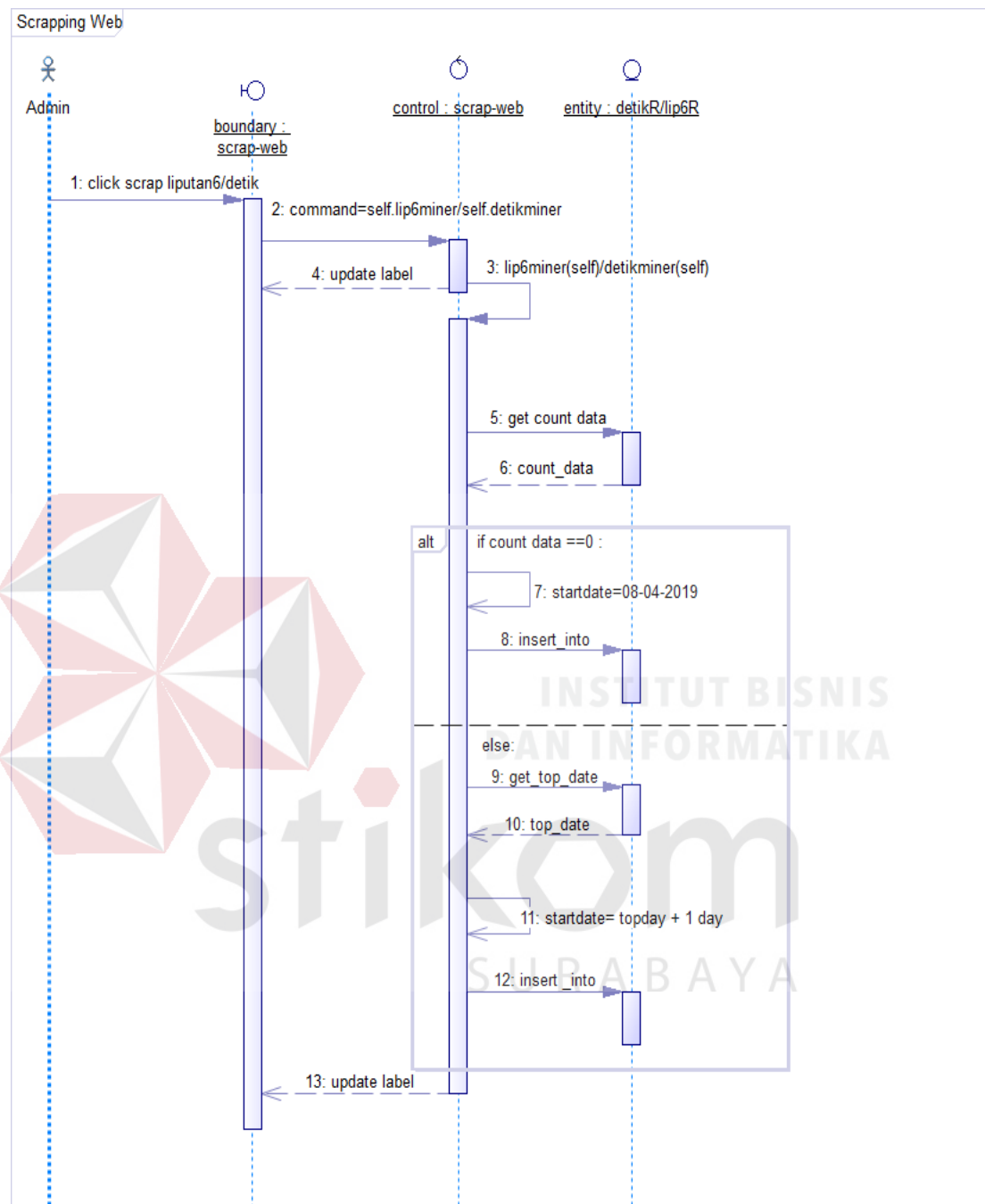
B.1 . *Sequence Diagram Login*



Gambar 3.10 Sequence Diagram *Login*

Diagram diatas memiliki beberapa komponen yaitu pengguna(*actor*), *login(boundary)* *login(controller)*, dan *login(entity)*. Adapun alur dari aplikasi antara lain adalah ; aktor memasukkan username dan password. Kemudian informasi tersebut akan diteruskan sebagai variable *uname* dan *psswd* kepada *controller*. Controller akan menginisiasi fungsi *login* yang kemudian akan mengecek *database* apakah ada *role* yang sesuai dengan password dan username yang dimasukkan. Halaman yang dibuka aplikasi akan bergantung terhadap *role* yang dimasukkan aktor. Jika *role* tidak ditemukan aplikasi akan memunculkan popup peringatan.

B.2. Sequence Diagram Scrapping Web

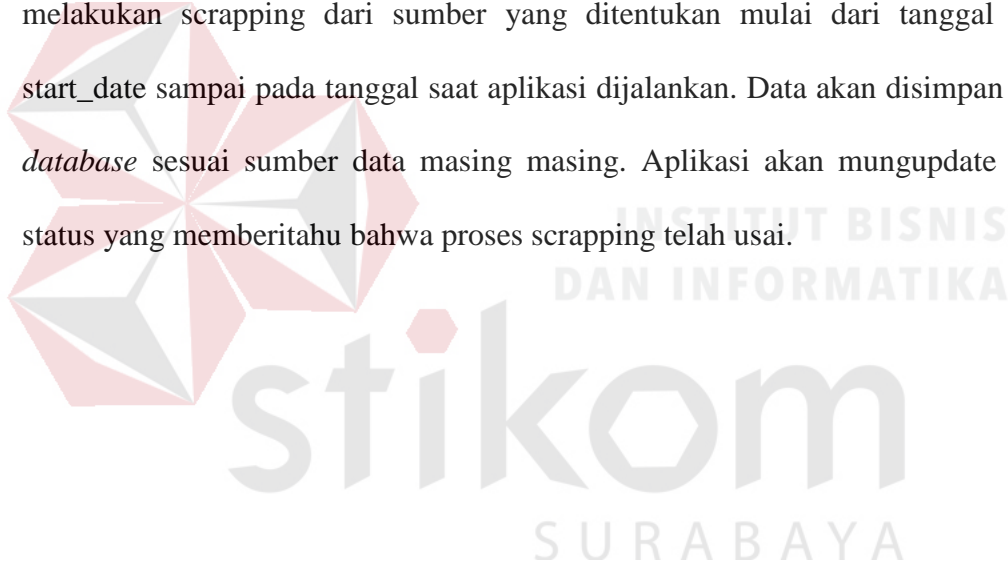


Gambar 3.11 Sequence Diagram *Scrapping Web*

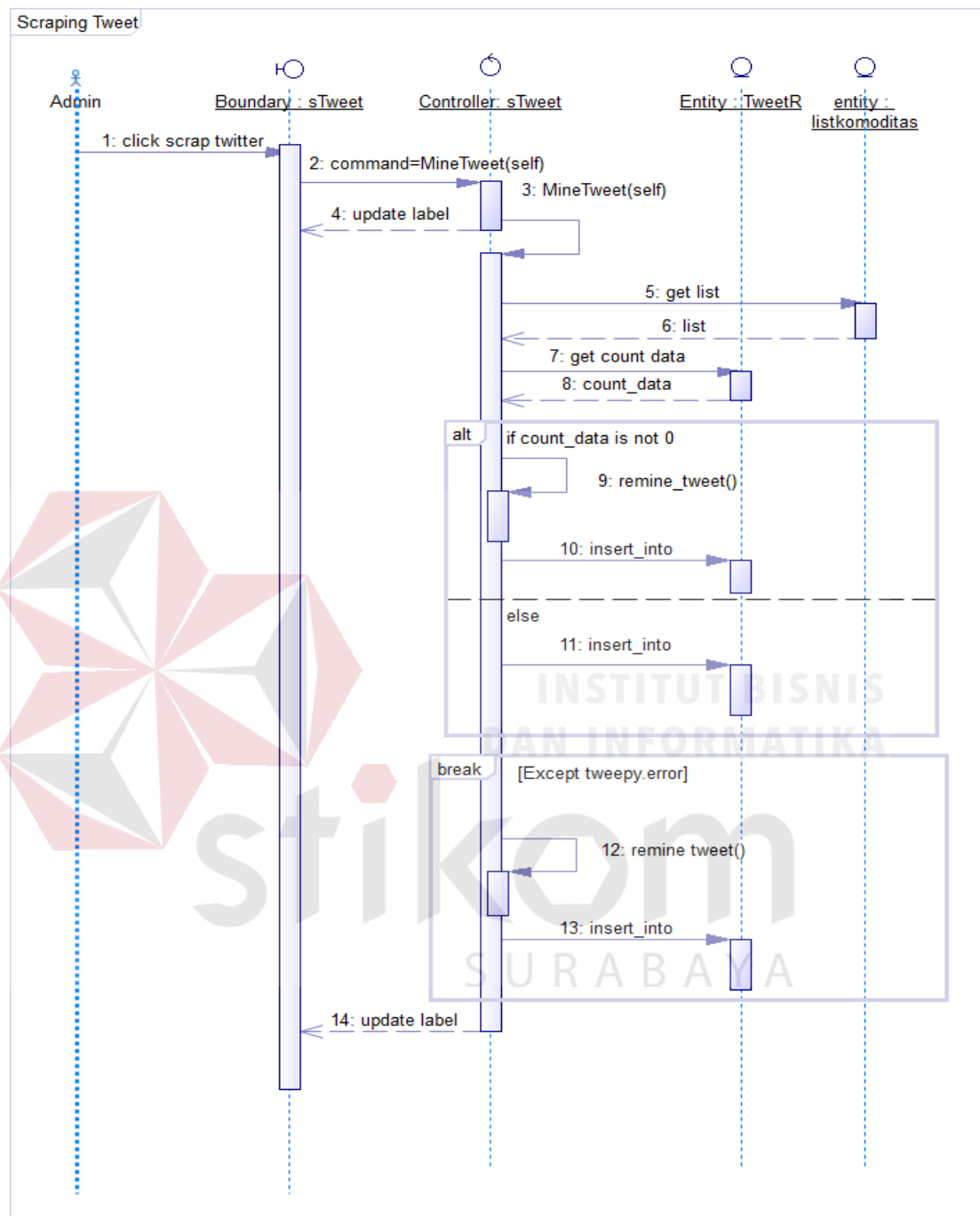
Pada diagram diatas terdapat 5 komponen yaitu ; admin(aktor), scrap-web(*boundary*), scrap-web (*control*), dan detikR/lip6R(*entity*). *Sequence* diagram ini menggambarkan proses *scrapping* untuk liputan 6 dan detik yang prosesnya sama namun dengan perbedaan sumber data yang diambil dan sumber data

penyimpanan. Adapun alur dari diagram tersebut antara lain adalah; aktor memilih tombol *scrap* liputan6 atau *scrap* detik. Fungsi *lip6miner(self)* akan diinisiasi untuk *scrapping* liputan6 dan fungsi *detikminer(self)* akan diinisiasi untuk *scrapping* Detik. Aplikasi akan mengupdate label status yang menandakan bahwa proses *scrapping* sedang dimulai.

Fungsi melakukan pengecekan tanggal. Jika data tanggal kosong aplikasi akan menetapkan *start_date* tanggal 8 april 2019 namun jika ada aplikasi akan menetapkan *start_date* tanggal teratas ditambah satu hari. Kemudian aplikasi akan melakukan *scrapping* dari sumber yang ditentukan mulai dari tanggal pada *start_date* sampai pada tanggal saat aplikasi dijalankan. Data akan disimpan pada *database* sesuai sumber data masing masing. Aplikasi akan mengupdate label status yang memberitahu bahwa proses *scrapping* telah usai.



B.3 . Sequence Diagram *Scrapping* Twitter



Gambar 3.12 Sequence Diagram *Scrapping* Twitter

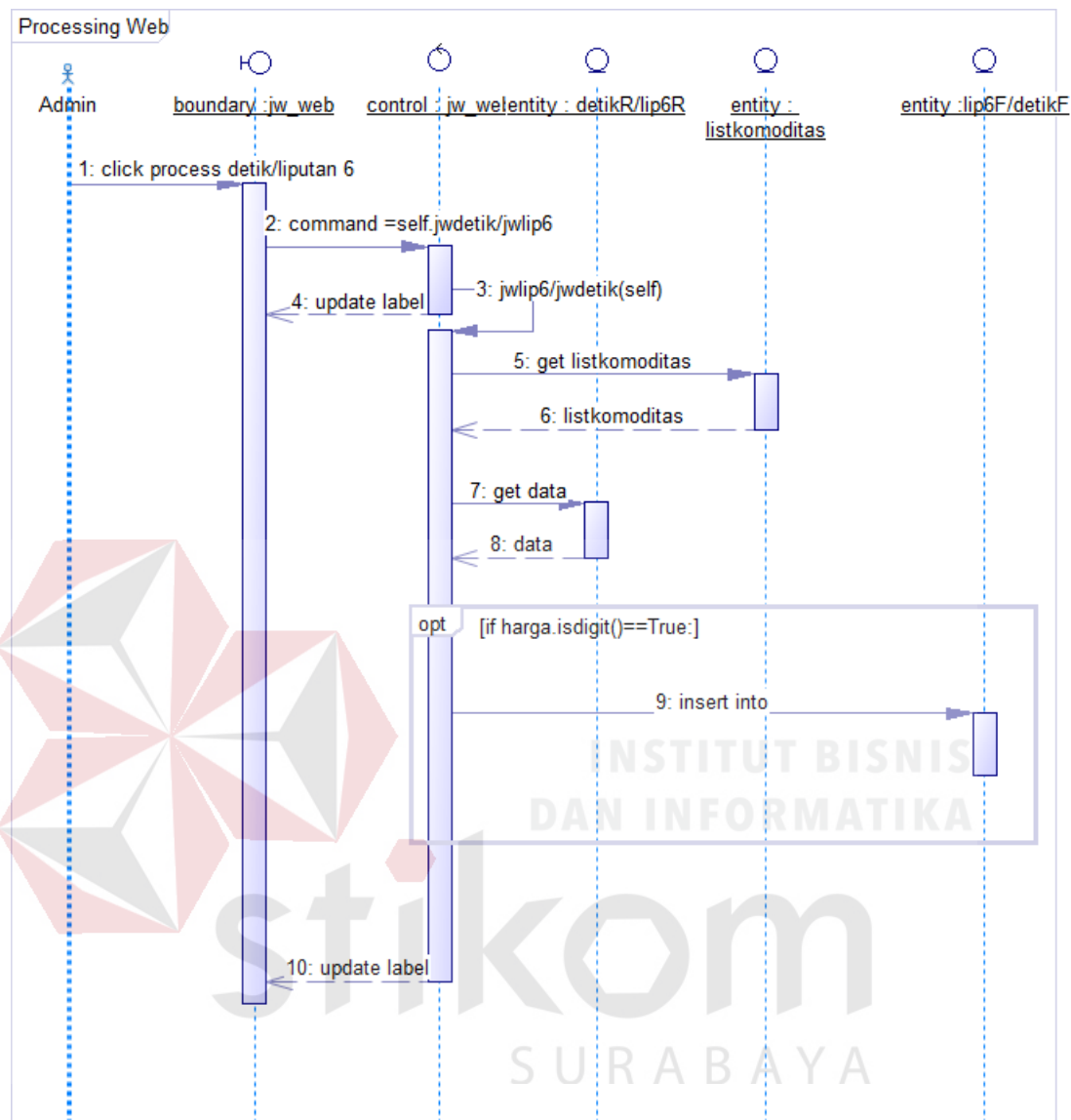
Pada diagram diatas terdapat 5 komponen yaitu ; admin(aktor), *sTweet(boundary)*, *sTweet (control)*, dan *TweetR(entity)*. Adapun alur dari diagram tersebut antara lain adalah; aktor memilih tombol *scrap* twitter. Fungsi

MineTweet(self) akan diinisiasi untuk *scrapping* Twitter. Aplikasi akan mengupdate label status yang menandakan bahwa proses *scrapping* sedang dimulai.

Aplikasi akan mengambil list komoditas yang merupakan list dua dimensi yang berisi keyword pencarian dan jenis komoditas. Selanjutnya aplikasi akan melakukan pengecekan tanggal. Jika data tanggal kosong aplikasi akan melakukan *scrapping* data seminggu sejak tanggal aplikasi dijalankan, dengan menggunakan keyword yang ada di list komoditas dan dilakukan sebanyak jumlah data yang ada di list komoditas,

Namun jika data kosong aplikasi akan menginisiasi fungsi *remine_tweet(kmdts)* yang memiliki proses yang sama dengan fungsi MineTweet(self) tetapi *scrapping* dilakukan sejak satu hari dari data tanggal terakhir didalam *database* . Data *scrapping* akan disimpan pada *database*. Aplikasi akan mengupdate label status yang memberitahu bahwa proses *scrapping* telah usai. Karena adanya pembatasan banyak data yang dapat diambil dari pihak Twitter, maka jika batas pengambilan data terlampaui, aplikasi akan memenuhi waktu tunggu dan mengupdate label status. Setelah itu aplikasi akan menjalankan fungsi *remine_tweet(kmdts)*

B.4. Sequence Diagram Web Processing



Gambar 3.13 Sequence Diagram Web *Processing*

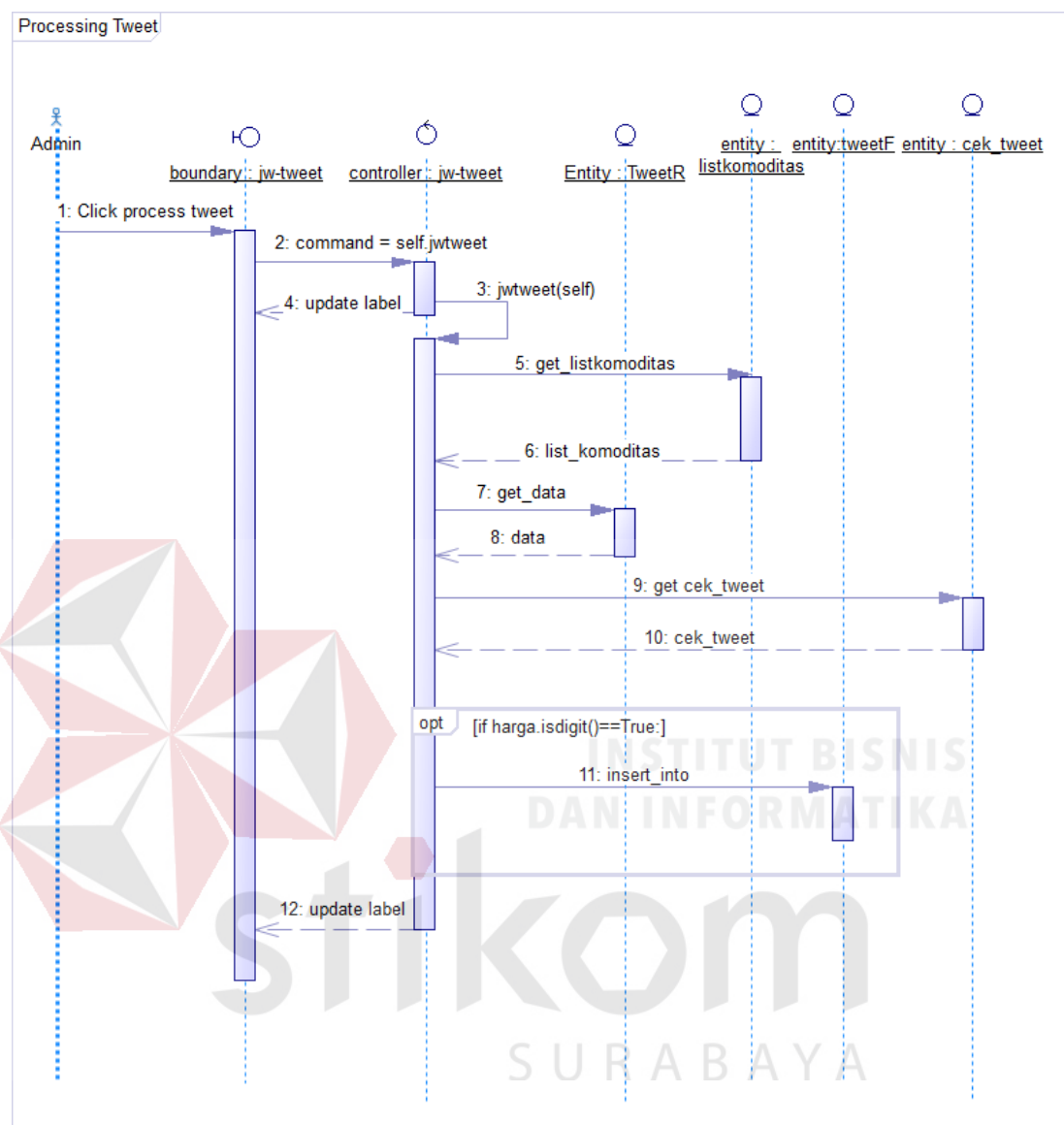
Pada diagram diatas terdapat 6 komponen yaitu ; admin(aktor), jw-web(*boundary*), jw-web (*control*),listkomoditas(*entity*),detikF/lip6F(*entity*) dan detikR/lip6R(*entity*). *Sequence* diagram ini menggambarkan alur memproses data hasil *scrapping* untuk liputan 6 dan Detik yang prosesnya sama namun dengan perbedaan sumber data yang diambil dan sumber data penyimpanan. Adapun alur dari diagram tersebut antara lain adalah; aktor memilih tombol *process* liputan6

atau *process* Detik. Fungsi *jwlip6* (self) akan diinisiasi untuk memproses hasil *scrapping* dari *Liputan6* dan fungsi *detikminer*(self) akan diinisiasi untuk memproses hasil *scrapping* dari detik. Aplikasi akan mengupdate label status yang menandakan bahwa pemrosesan data sedang dimulai.

Aplikasi akan mengambil list komoditas yang berisi keyword pencarian dan jenis komoditas. Lalu aplikasi akan mengambil semua data berita dari sumber data. Data berita tersebut akan melalui proses pembersihan tag, *case folding*, *filtering*, dan *tokenizing* perkalimat. Selanjutnya data list komoditas juga akan melalui proses *tokenizing* perkata yang dimasukkan kedalam list lkomoditas. Untuk setiap data hasil *tokenizing* perkalimat berita tadi akan di *tokenize* kembali perkata dan dimasukkan ke dalam list dua dimensi *word* yang berisi nomor urutan dalam kalimat dan hasil *tokenize* kata.

Untuk setiap list *word* akan dilakukan pengecekan algoritma jaro-winkler dengan list lkomoditas. Jika kata pertama di lkomoditas ditemukan di *word* maka pencarian akan berlanjut pada kata kedua lkomoditas dan kata selanjutnya di *word* sesuai dengan jumlah kata di lkomoditas. Jika semua kata ditemukan maka aplikasi akan melakukan pengecekan algoritma jaro-winkler didalam *word* terhadap kata “rp” untuk menemukan harga komoditas. Jika harga ditemukan aplikasi akan melakukan pengecekan kembali terhadap data pada tanggal tersebut. Jika terdapat data, aplikasi akan melakukan perhitungan rata-rata, dan mengupdate data didalam sumber data *lip6F* atau *detikF*. Apabila data tidak ditemukan maka harga akan langsung disimpan didalam *database*. Setelah itu, aplikasi akan mengupdate label yang menandakan proses telah selesai

B.5. Sequence Diagram *Tweet Processing*



Gambar 3.14 Sequence Diagram *Tweet Processing*

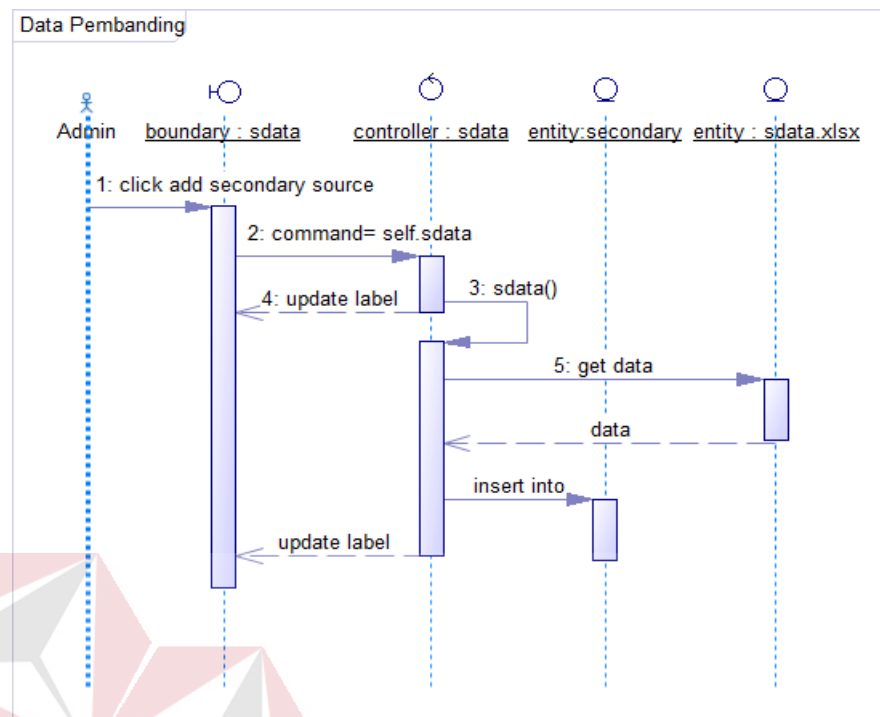
Pada diagram diatas terdapat 6 komponen yaitu ; admin(aktor), *jw-tweet*(boundary), *jw-tweet* (control), *listkomoditas*(entity), *tweetF*(entity), *cek_tweet*(entity) dan *tweetR*(entity). *Sequence* diagram ini menggambarkan alur memproses data hasil *scrapping* untuk twitter. Adapun alur dari diagram tersebut antara lain adalah; aktor memilih tombol *process* twitter. Fungsi *jwtweet* (self)

akan Aplikasi akan mengupdate label status yang menandakan bahwa pemrosesan data sedang dimulai.

Aplikasi akan mengambil list komoditas yang berisi keyword pencarian dan jenis komoditas. Lalu aplikasi akan mengambil semua data berita dari sumber data. Data berita tersebut akan melalui proses pembersihan tag, *case folding*, *filtering*, dan *tokenizing* perkata dan dimasukkan ke dalam list dua dimensi *word* yang berisi nomor urutan dalam kalimat dan hasil tokenize kata. Selanjutnya data list komoditas juga akan melalui proses *tokenizing* perkata yang dimasukkan kedalam list lkomoditas.

Untuk setiap list *word* akan dilakukan pengecekan algoritma jaro-winkler dengan list lkomoditas. Jika kata pertama di lkomoditas ditemukan di *word* maka pencarian akan berlanjut pada kata kedua lkomoditas dan kata selanjutnya di *word* sesuai dengan jumlah kata di lkomoditas. Jika semua kata ditemukan maka aplikasi akan melakukan pengecekan algoritma jaro-winkler didalam *word* dengan kata yang ada didalam *database cek_tweet*. Jika semua kata tidak ditemukan maka pengecekan jaro-winkler terhadap kata “rp” untuk menemukan harga komoditas akan dilakukan. Jika harga ditemukan aplikasi akan melakukan pengecekan kembali terhadap data pada tanggal tersebut. Jika terdapat data, aplikasi akan melakukan perhitungan rata-rata, dan mengupdate data didalam sumber data lip6F atau detikF. Apabila data tidak ditemukan maka harga akan langsung disimpan didalam *database*. Setelah itu, aplikasi akan mengupdate label yang menandakan proses telah selesai

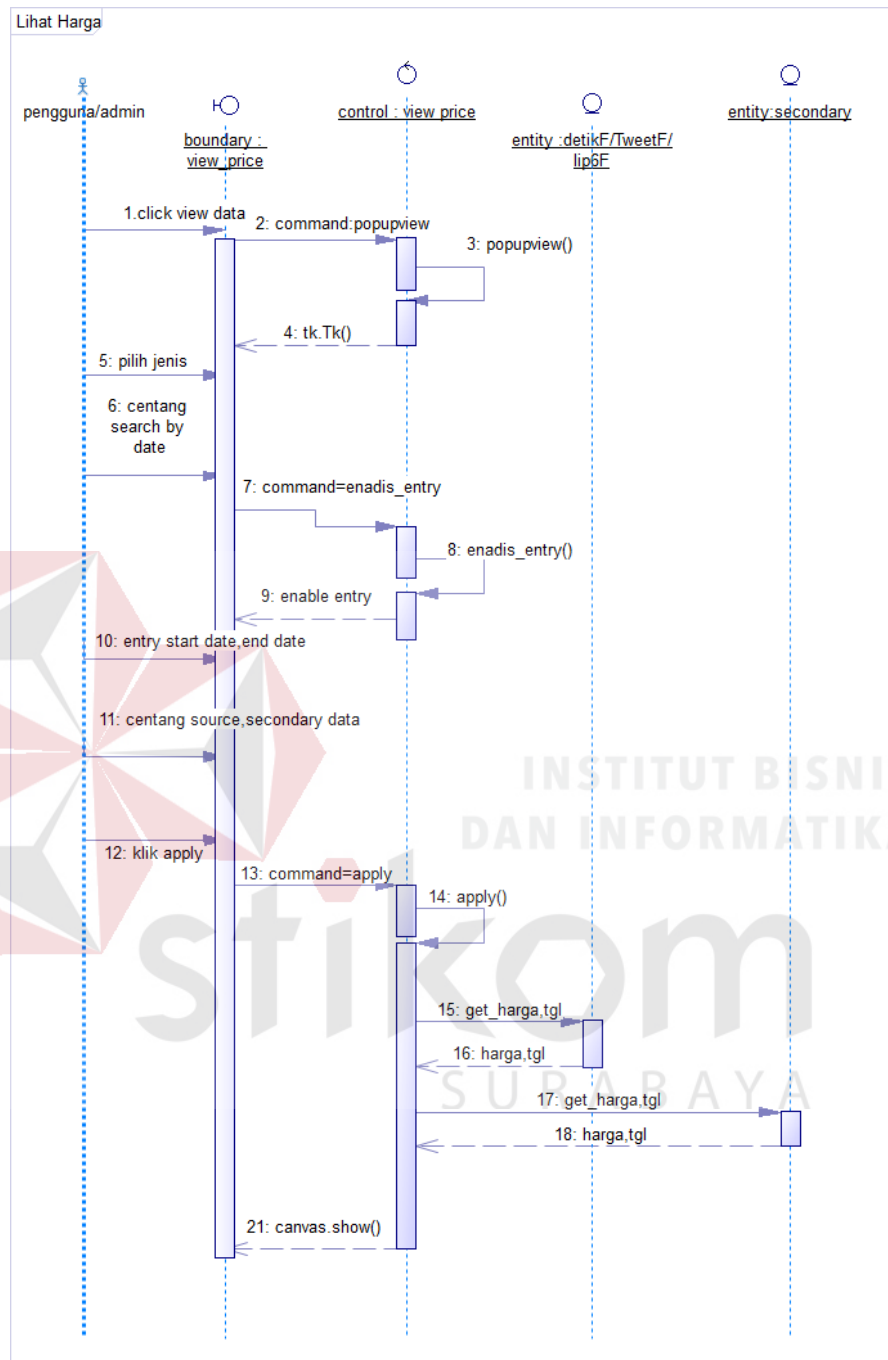
B.6. Sequence Diagram *Import Data Pembanding*



Gambar 3.15 Sequence Diagram *Import Data Pembanding*

Pada diagram diatas terdapat 6 komponen yaitu ; admin(aktor), sdata(boundary), sdata (control), source(entity), sdata.xlsx(entity). Setelah pengguna memilih tombol *add secondary source*, label status akan diupdate dan aplikasi akan menginisiasi fungsi sdata() yang akan mengambil data dari sumber sdata.xlsx dan memasukkannya kedalam database *source*. Label status akan diupdate untuk menandakan proses telah selesai

B.7. Sequence Diagram Lihat Harga



Gambar 3.16 Sequence Diagram Lihat Harga

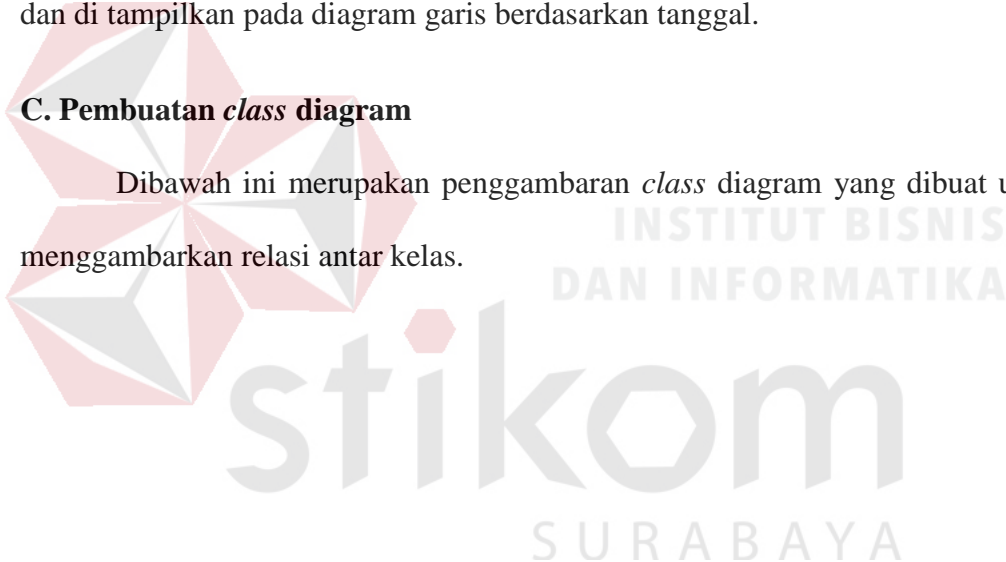
Pada diagram diatas terdapat 6 komponen yaitu ; pengguna/admin(aktor), secondary(*boundary*), sdata (*control*), source(entity), *tweetF/lip6F/detikF*(entity). *Sequence diagram* ini menggambarkan proses untuk melihat data harga yang dapat

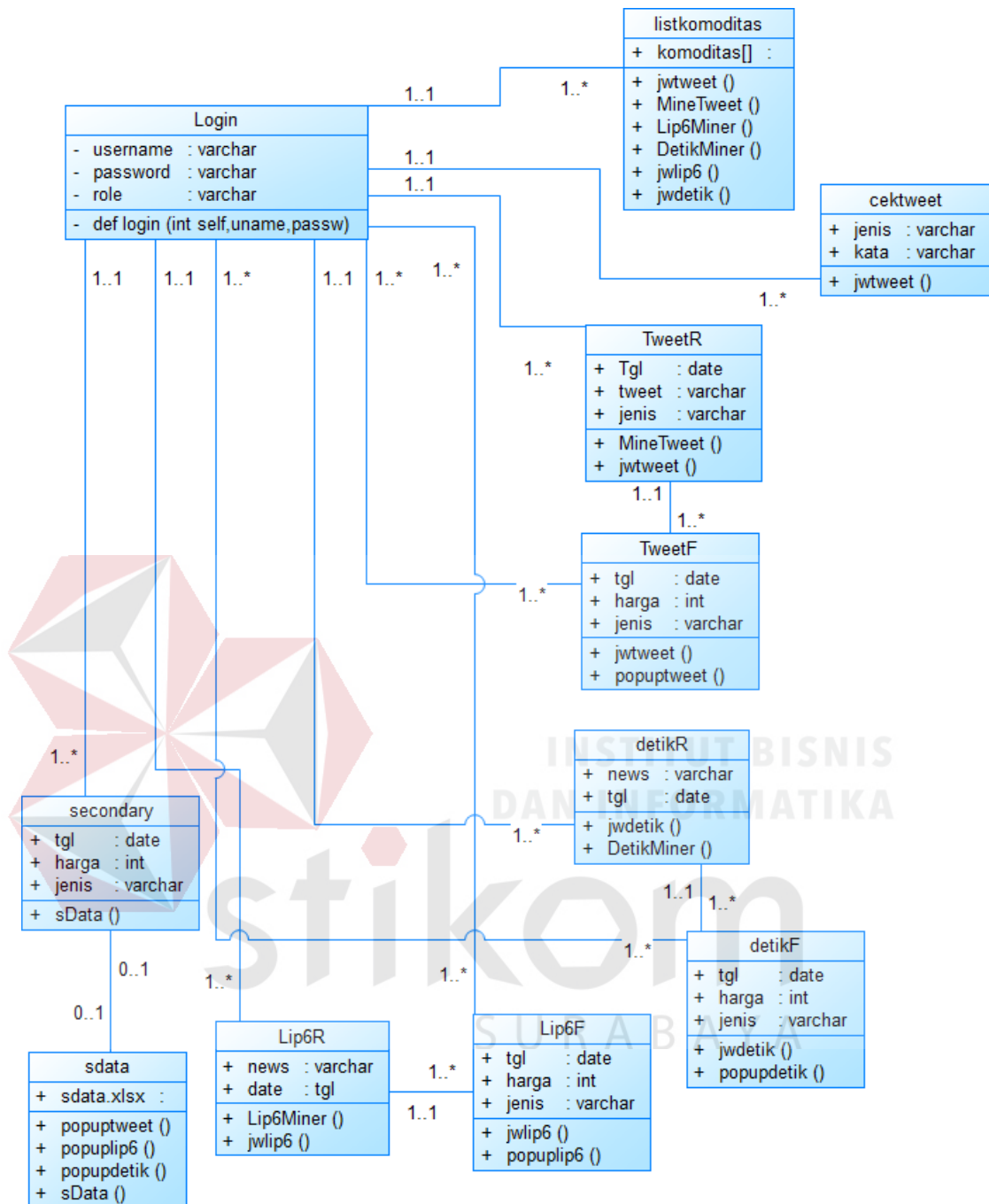
dilakukan oleh pengguna maupun admin, keduanya memiliki alur proses yang sama namun dengan hak akses yang berbeda.

Proses dimulai saat aktor milih tombol *view data*. Tombol tersebut akan melaksanakan Fungsi *popupviewdata()*. Fungsi tersebut akan menampilkan halaman popup untuk melihat data harga. Aktor akan memilih jenis dan dapat mencentang sumber data pilihan yaitu Twitter, Liputan6, Secondary, dan detik. Untuk mencari berdasarkan tanggal actor bisa mencentang *search by date* dan memasukkan data tanggal mulai dan berakhir. Data akan diambil dari sumber data dan di tampilkan pada diagram garis berdasarkan tanggal.

C. Pembuatan *class diagram*

Dibawah ini merupakan penggambaran *class diagram* yang dibuat untuk menggambarkan relasi antar kelas.



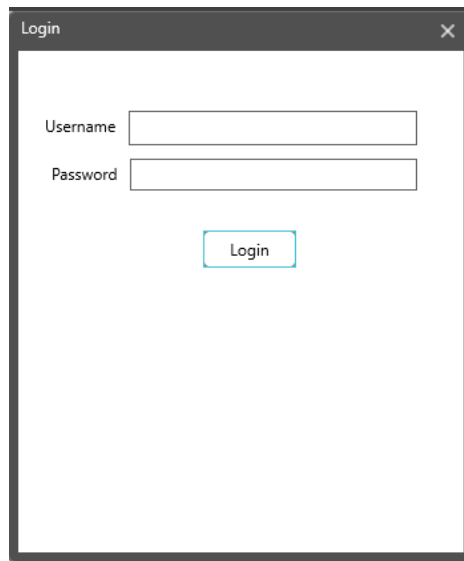


Gambar 3.17 Class Diagram

3.3.3 Desain User Interface (UI)

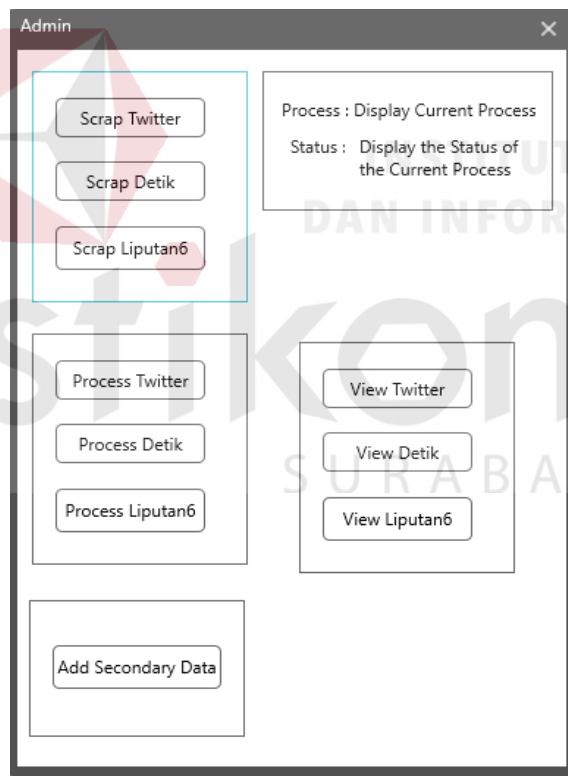
Berikut merupakan gambar desain antarmuka yang akan diterapkan dalam aplikasi

:



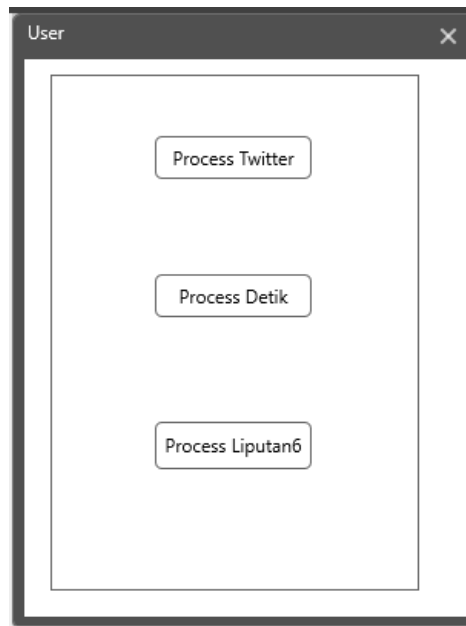
A screenshot of a 'Login' window. It features a title bar with the text 'Login' and a close button. The window contains two input fields: 'Username' and 'Password'. Below these fields is a 'Login' button. The window has a dark border and a light background.

Gambar 3.18 Desain Halaman *Login*

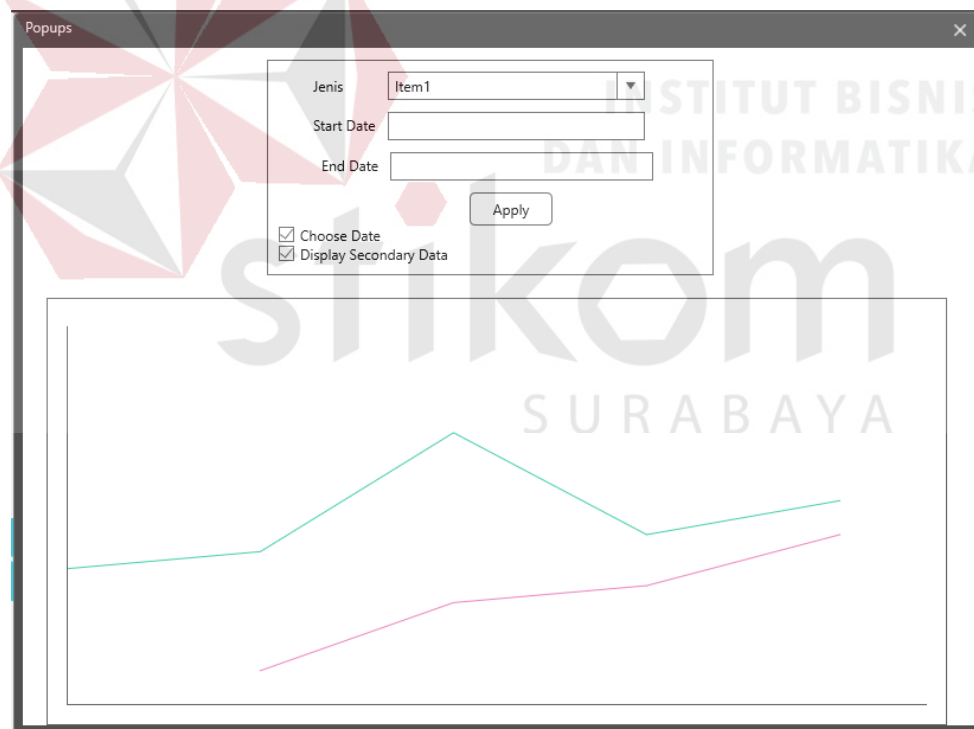


A screenshot of an 'Admin' window. It features a title bar with the text 'Admin' and a close button. The window is divided into several sections. On the left, there is a vertical list of buttons: 'Scrap Twitter', 'Scrap Detik', and 'Scrap Liputan6'. Below these is a section with buttons: 'Process Twitter', 'Process Detik', and 'Process Liputan6'. At the bottom left is a button labeled 'Add Secondary Data'. On the right, there is a box containing the text 'Process : Display Current Process' and 'Status : Display the Status of the Current Process'. Below this box is another section with buttons: 'View Twitter', 'View Detik', and 'View Liputan6'. The window has a dark border and a light background.

Gambar 3.19 Desain Halaman Admin



Gambar 3.20 Desain Halaman Pengguna




Gambar 3.21 Desain Halaman *Popup* Lihat Harga

3.4. Construction

3.4.1 Penerapan Text Mining

Pada tahap ini aplikasi akan dibangun sesuai dengan permodelan yang telah dibuat. Aplikasi akan ditulis menggunakan bahasa pemrograman *Python* 3.4 dan menggunakan DBMS Microsoft SQL Server 2017. Berikut merupakan penerapan proses *textmining* beserta kodenya didalam aplikasi. Dalam penjelasan berikut beberapa proses akan dibagi berdasarkan sumber Twitter dan Web. Untuk sumber web merangkum proses dari text mining pada liputan6 dan detikcom, karena proses dari kedua sumber tersebut sama persis hanya diambil dari sumber yang berbeda.

Pembersihan tag



```

"""Proses Pembersihan Tag"""
tex=tex.replace("googletag.cmd.push"," ")
#replace(kata yang dihapus,kata yang diganti)
tex=tex.replace("(","").replace(")","").replace("{","").replace("}", "")
tex=tex.replace("function "," ").replace("googletag.display"," ")
tex=tex.replace("'div-gpt-ad-1535944306169-0'", " ")

```

Gambar 3.22 Kode Pembersihan Tag Web

Pembersihan tag merupakan tahap pertama dalam pemrosesan data. Gambar di atas merupakan kode yang digunakan untuk membersihkan tag didalam web. Tag-tag ini lebih sering ditemukan pada sumber detik.com. Adapun pembersihan tag dalam twitter dilakukan dengan kode dibawah ini.

Pada tweet bentuk-bentuk tag yang dibersihkan adalah *emoticon*, *Uniform Resource Locator (URL)*, *hashtag(#)* dan *mention* dengan karakter *@*. Pembersihan *emoticon* dilakukan dengan menggunakan list *emotes* yang berisi kode-kode *emoticon* dalam *tweet*.


```

#list kode emoticon
emotes=["\\xf0\\x9f\\x8d\\x96", "\\xf0\\x9f\\x98\\xad", "\\xf0\\
#mengambil tweet di db
for tweet in CClean:
    rtweet.append(tweet)
#pada tiap tweet di db akan dilakukan proses
for tweet in rtweet:
    date=tweet[0]
    t=tweet[1]
    jenis=tweet[2]
    tweet=BeautifulSoup(t, 'lxml')
    tweet=tweet.get_text()
    # setiap kode emoticon di list di cek di tweet
    for i in emotes:
        #jika ada dihilangkan
        tweet=tweet.replace(i, " ")
    # menghapus hashtag
    tweet = re.sub(r':\#[^\w_]+[\w\'\_-\]*[\w_]+', '', tweet)
    # menghapus newline
    tweet=tweet.replace(r"\n", " ")
    # Menghapus URL
    tweet=re.sub(r"(?:\@|https?\:\/\/)\s+", "", tweet)
    # menghapus @ mention
    tweet=re.sub(r'@[A-Za-z0-9]+', '', tweet)
    # menghapus notasi selain huruf
    tweet=re.sub('[^A-Za-z0-9]+', ' ', tweet) [2:]

```

Gambar 3.23 Kode Pembersihan Tag Twitter

Casefolding

Selanjutnya adalah proses *casefolding* yang akan merubah semua huruf menjadi huruf kecil. Penerapan kode yang digunakan pada twitter dan web menggunakan fungsi yang sama yaitu “.lower()” seperti yang ditunjukkan pada gambar berikut. Variabel “tweet” merupakan data yang akan diproses.

```

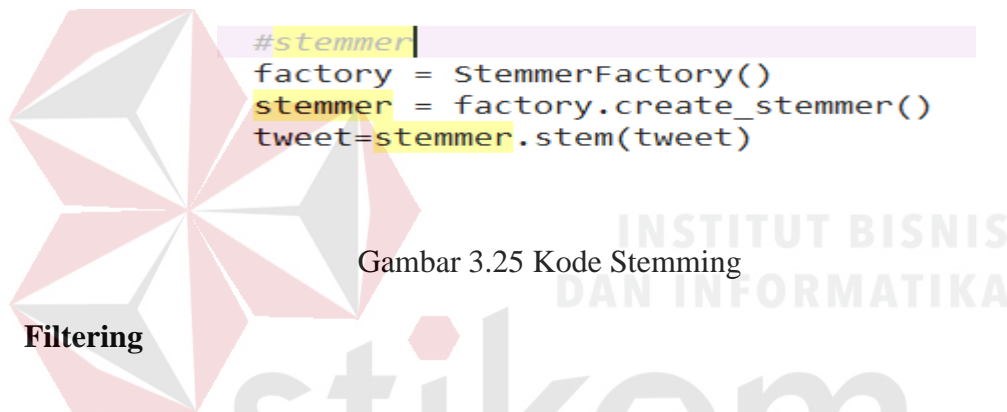
#casefolding
tweet=tweet.lower()

```

Gambar 3.24 Kode Casefolding

Stemming

Pada tahap *stemming*, data akan dirubah menjadi kata dasar. Contoh kata “mencuri” akan dirubah menjadi “curi”. Pada tahap ini penulis menggunakan modul Pysastrawi. Penerapannya modul ini pada sumber web dan twitter sama persis. *Stemmer* dalam modul akan mengecek apakah string memiliki kata imbuhan. Jika iya maka kata imbuhan tersebut akan dihilangkan. Untuk lebih detilnya dapat dilihat pada gambar berikut ini. Variabel “*tex*” merupakan data yang akan diproses.



Filtering

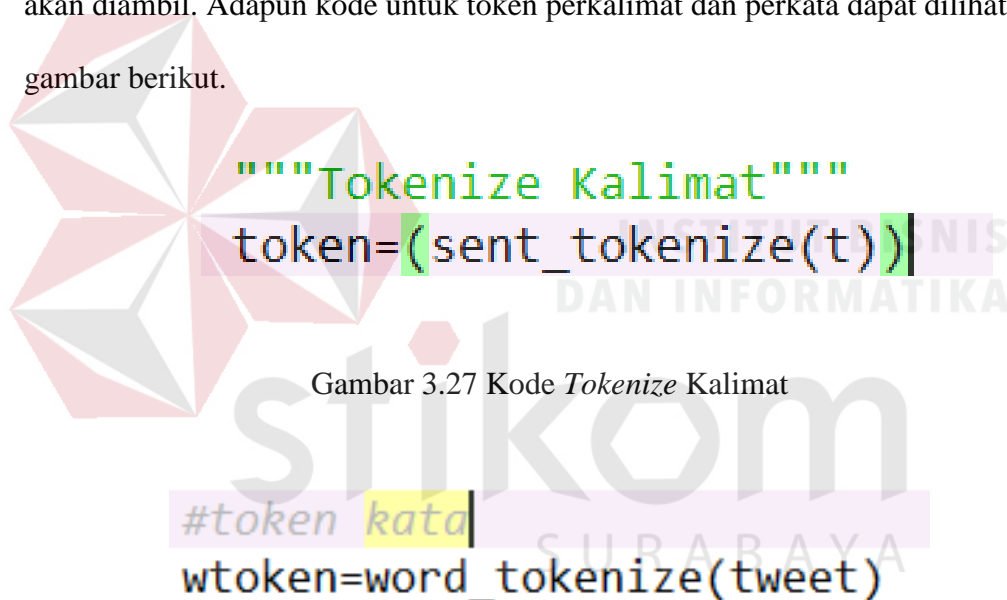
Pada tahap *filtering* penulis akan kembali menggunakan modul Pysastrawi. Modul filtering akan mengecek apakah terdapat kata-kata khusus yang dianggap tidak penting. Jika tidak ada, maka kata tersebut akan dihilangkan. Adapun penggunaan dalam sumber web dan twitter sama persis, lebih jelasnya dapat dilihat pada gambar dibawah ini.

```
#stopword
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
tweet = stopword.remove(tweet)
```

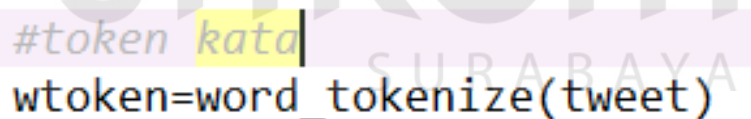
Gambar 3.26 Kode *Filtering*

Tokenizing

Untuk proses *tokenizing* pada sumber web akan dilakukan dua kali yaitu perkata dan perkalimat dikarenakan data yang banyak. Hal ini dilakukan agar data harga dapat dicari perkalimat. Sedangkan untuk twitter hanya dilakukan perkata saja. Dalam implementasinya *tokenizing* berperan penting dalam seleksi jaro winkler sehingga akan dijelaskan lebih lanjut pada tahap selanjutnya. Untuk *tokenizing* perkata, token diambil menggunakan acuan spasi antar kata. Sedangkan *tokenizing* perkalimat, token diambil dengan acuan titik dan spasi. Contoh : "Daging Sapi. Menurut". Jika terdapat titik dan ada spasi setelahnya maka token akan diambil. Adapun kode untuk token perkalimat dan perkata dapat dilihat pada gambar berikut.



Gambar 3.27 Kode *Tokenize* Kalimat



Gambar 3.28 Kode *Tokenize* Kata

Seleksi Jaro-Winkler

Seleksi Jaro Winkler dilakukan dengan modul pyjarowinkler. Pada proses penyeleksian jaro-winkler untuk sumber web dan Twitter terdapat beberapa perbedaan. Untuk sumber web, penyeleksian dilakukan pada token kalimat dari data berita untuk mempermudah dan meningkatkan akurasi penelitian sehingga

dibutuhkan, sedangkan twitter langsung dilakukan pada data *tweet*. Lalu pada twitter dilakukan pengecekan jaro-winkler dengan menggunakan *database* “cek_tweet” untuk mengetahui apa *tweet* relevan atau tidak.

Dalam penyeleksian jaro-winkler terdapat *list* komoditas yang berisi nama komoditas dan jenis komoditas. Berikut merupakan contoh beberapa komoditas dalam *list*.

```
#isi list[nama yang akan dicari,jenisnya]
komoditas=[["bayam","bayam"],
            ["daging ayam broiler","daging ayam ras"]]
```

Gambar 3.29 List Komoditas

Nama komoditas didalam *list* komoditas akan dipecah perkata menggunakan *tokenize* kata dan disimpan pada *list* lkomoditas. Kode dapat dilihat pada gambar berikut.

```
#dilakukan perkomoditas
for a in komoditas:
    #komoditas dipecah perkata
    tkomoditas=word_tokenize(a[0])
    lkomoditas.append(tkomoditas)
```

Gambar 3.30 Kode Pemecahan Kata Komoditas

Proses selanjutnya merupakan perbedaan pertama antar sumber yang sudah dijelaskan tadi. Proses penyeleksian akan terjadi tiap token kalimat untuk web yang ditandai dengan inisiasi *tokenizing* kata. Untuk sumber Twitter *tokenizing* akan langsung di inisiasi tanpa token kalimat. Berikut adalah gambar *tokenizing* kalimat dan kata pada web. Token kata disimpan pada *list* wtoken

```
#tiap kalimat akan dilakukan proses berikut
for b in token:
    #tokenizing kata di tiap kalimat
    wtoken=word_tokenize(b)
```

Gambar 3.31 Awal Proses Seleksi Pada Web

Semua proses selanjutnya dilakukan didalam perulangan pada gambar diatas. Setelah itu *list* wtoken akan diberi penomoran untuk mengetahui urutannya dalam kalimat ataupun data *tweet* dan data tersebut akan disimpan pada *list words*. Untuk kodenya dapat dilihat pada gambar berikut.

```
#tiap token kata
for c in wtoken:
    i=i+1
    #isi list word[nomor urut,kata]
    words.append([i,c])
```

Gambar 3.32 Kode Penomoran Kata

Selanjutnya akan dilakukan pencarian kata pertama pada nama komoditas didalam data/kalimat menggunakan algoritma jaro-winkler. Jika seandainya nama komoditas adalah “daging sapi” maka kata daging akan dicari didalam data/kalimat. Jika ditemukan kata dengan tingkat persamaan 0,90 atau 90% maka pencarian akan di teruskan ke kata selanjutnya dalam nama komoditas. Dibawah ini merupakan gambar kode pencarian kata pertama nama komoditas.

```

for d in words:
    x=0
    #pencarian komoditas dilakukan pada tiap kata
    for a in lkomoditas:
        y=1
        z=0
        #pencarian nama komoditas pertama pada kata pertama
        jw=distance.get_jaro_distance(a[0],d[1], winkler=True, scaling=0.1)
        #jika kesamaan 90% akan dilakukan pencarian selanjutnya
        if jw>=0.90:

```

Gambar 3.33 Kode Pencarian Kata Pertama Nama Komoditas

Setelah kata nama komoditas pertama ditemukan dalam berita/tweet (nw), kata setelah nama komoditas dalam berita (nw) tersebut akan diambil untuk dicek dengan kata nama komoditas selanjutnya, jika nama komoditas daging sapi, maka kata sapi akan dibandingkan dengan kata setelah nama komoditas dalam berita (nw). Proses ini akan berulang sebanyak sisa kata yang ada didalam nama komoditas.

```

for d1 in words:
    if d1[0]==x1+x2:
        #kata setelah nama komoditas pertama yang ditemukan pada data berita/tweet
        nw=d1[1]
        #pencarian sisa nama komoditas pada kalimat/data
        while y<=len(a)-1:
            jw=distance.get_jaro_distance(a[y],nw, winkler=True, scaling=0.1)
            if jw>=0.90:
                z+=1
                x2+=1
                nw2=nw
                for d1 in words:
                    if d1[0]==x1+x2:
                        nw=d1[1]
            y+=1

```

Gambar 3.34 Pencarian Sisa Kata Nama Komoditas Didalam Berita.

Proses selanjutnya adalah perbedaan kedua antar sumber yang sudah disebutkan pada awal proses seleksi jaro winkler. Proses ini khusus dilakukan didalam twitter yaitu pengecekan kata-kata yang menentukan apakah tweet tersebut relevan atau tidak menggunakan *database* “cek_tweet”. Setiap kata didalam database akan dicocokkan dengan tiap kata pada kalimat berita/*tweet*. Jika kata tidak ditemukan maka proses seleksi akan dilanjutkan.

```
#kata didalam database diambil
quer="Select kata From cek_tweet where Jenis =?"
param=jenis
Ckata.execute(quer,param)
lcekTweet=Ckata.fetchall()
x4=0
#setiap kata didalam DB akan dicocokkan dengan tiap kata pada kalimat/tweet
for row in lcekTweet:
    a=', '.join(row)
    for x in words:
        jw=distance.get_jaro_distance(a,x[1], winkler=True, scaling=0.1)
        if jw==1.00:
            x4+=1
#jika kata tidak ditemukan proses berlanjut
if x4==0:
```

Gambar 3.35 Kode Penggunaan Database “cek_tweet”

Setelah semua nama komoditas ditemukan dalam berita, harga dari komoditas tersebut akan dicari. Adapun pencariannya akan dilakukan setelah kata nama komoditas terakhir ditemukan. Pencarian harga pertama kali dilakukan dengan mencari kata “Rp” dalam berita/*tweet*. Jika kata “Rp” ditemukan maka lokasi kata tersebut akan disimpan sehingga jika ada dua atau lebih komoditas dalam satu kalimat atau *tweet*, tempat tersebut tidak akan diperiksa lagi untuk kata “Rp”. Kemudian lokasi setelah ditemukannya kata “Rp” akan disimpan untuk

proses selanjutnya. Untuk lebih jelasnya, proses dapat dilihat pada gambar dibawah ini.

```
for x2 in words:
    if x2[1]==nw2:
        #tempat nama komoditas terakhir ditemukan
        x5=x2[0]
for x2 in words:
    #harga dicari setelah tempat nama komoditas terakhir ditemukan
    if x2[0]>x5:
        if x4==0:
            #pencarian harga dengan rp
            jw=distance.get_jaro_distance("Rp",x2[1], winkler=True, scaling=0.1)
            if jw==1.00:
                if not lprice:
                    #Lokasi Rp ditemukan disimpan.
                    lprice.append(x2[0])
                    #Lokasi setelah kata Rp
                    z=x2[0]+1
```

Gambar 3.36 Kode Pencarian kata Rp.

Kemudian data harga pada lokasi setelah kata “Rp.” akan diambil. Selanjutnya akan dicek apakah ada data harga pada tanggal yang sama pada database. Jika tidak ada data akan disimpan kedalam database.

```
#pengecekan apa data harga ada di database
quer="select * from testWF where jenis=? and tgl=? "
param=jenis,date
Ckata.execute(quer,param)
ldata=ckata.fetchall()
HFinal=0
#jika tidak ada langsung disimpan
if not ldata:

    #data harga
    harga=x3[1]
    #penghilangan /kg dalam satu kata
    harga=harga.replace(".", "").replace("/kg", "")
    if harga.isdigit()==True:
        HFinal=int(harga)
        quer="insert into testWF(tgl,harga,jenis) values(?,?,?)"
        param=(date,HFinal,jenis)
        Ckata.execute(quer,param)
        #data disimpan
        ckata.commit()
```

Gambar 3.37 Kode penyimpanan harga komoditas.

Tetapi jika ditemukan harga komoditas pada tanggal yang sama maka harga yang ada di *database* akan ditambahkan dengan data yang baru dan dibagi dua untuk kemudahan visualisasi. Setelah itu data didalam *database* akan diupdate. Berikut merupakan kode dari proses tersebut.

```

else: #jika ada data harga pada tanggal yang sama
    for i in ldata:
        harga=x3[1]
        harga=harga.replace(".", "").replace("/kg", "")
        if harga.isdigit()==True:
            #harga dirata-rata
            HFinal=(i[1]+int(harga))/2
            print(HFinal)
            print(date)
            #harga database diupdate
            quer="update testWF set harga=? where tgl=? and jenis=?"
            param=(HFinal,date,jenis)
            ckata.execute(quer,param)
            ckata.commit()

```

Gambar 3.38 Kode Penghitungan Rata-Rata Harga

Untuk penerapan algoritma dalam penyeleksian data harga komoditas pangan murni hanya mencari nama komoditas dan harganya saja pada suatu teks. Sehingga faktor faktor lain seperti apakah komoditas tersebut adalah barang impor ataupun jika harga yang ada ditekst adalah target penyebaran pemerintah maupun data dari tahun lalu tidak diperiksa. Selain itu algoritma juga tidak memeriksa satuan dari harga apakah perkilo atau selusin. Algoritma juga akan menghitung rata-rata harga komoditas jika ditemukan ada lebih dari satu harga pada tanggal yang sama. Hal ini dilakukan karena untuk visualisasi data hanya pertanggal saja.

3.4.2 Testing

Pada tahap ini aplikasi yang telah melalui proses *coding* akan diuji menggunakan menggunakan skenario pengujian. Skenario pengujian sendiri dibagi menjadi dua yaitu uji coba fungsi, dan uji akurasi.

A. Skenario Uji Coba Fungsi

Berikut merupakan penyusunan skenario uji coba fungsi.

Tabel 3.2 Tabel Skenario Testing Fungsi

No	Fungsi	Input	Hasil yang diharapkan
1	Login Pengguna	User,user	Aplikasi menampilkan halaman pengguna
2	Login Admin	Admin,admin	Aplikasi menampilkan halaman admin
3	Scrapping	Menekan tombol <i>scrap</i> twitter	Aplikasi dapat mengambil data dari sumber dan menyimpan didalam <i>database</i> tweetR
		Menekan tombol <i>scrap</i> detik	Aplikasi dapat mengambil data dari sumber dan menyimpan didalam <i>database</i> detikR
		Menekan tombol <i>scrap</i> liputan 6	Aplikasi dapat mengambil data dari sumber dan menyimpan didalam <i>database</i> lip6R
4	Processing	Menekan tombol <i>process</i> twitter	Aplikasi dapat melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> tweetF

No	Fungsi	Input	Hasil yang diharapkan
		Menekan tombol <i>process</i> detik	Aplikasi dapat melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> detikF.
		Menekan tombol <i>process</i> liputan6	Aplikasi dapat melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> lip6F.
	Membersihkan Tag	Jakarta - Harga daging sapi lokal dan impor di Pasar Jatinegara mengalami kenaikan. Tercatat harga daging sapi lokal mencapai Rp 130.000/kg. Lalu, daging sapi beku impor dari India Rp 100.000/kg. Menurut Yusuf, salah satu pedagang daging sapi di Pasar Jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga. googletag.cmd.push	Jakarta - Harga daging sapi lokal dan impor di Pasar Jatinegara mengalami kenaikan. Tercatat harga daging sapi lokal mencapai Rp 130.000/kg. Lalu, daging sapi beku impor dari India Rp 100.000/kg. Menurut Yusuf, salah satu pedagang daging sapi di Pasar Jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.
	Case Folding	Jakarta - Harga daging sapi lokal dan impor di Pasar Jatinegara mengalami kenaikan. Tercatat harga daging sapi lokal mencapai Rp 130.000/kg. Lalu, daging sapi beku impor dari India Rp 100.000/kg. Menurut Yusuf, salah satu pedagang daging sapi di Pasar Jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.	Jakarta - harga daging sapi lokal dan impor di pasar jatinegara mengalami kenaikan. tercatat harga daging sapi lokal mencapai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. menurut yusuf, salah satu pedagang daging sapi di pasar jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.

No	Fungsi	Input	Hasil yang diharapkan
	Stemming	jakarta - harga daging sapi lokal dan impor di pasar jatinegara mengalami kenaikan. tercatat harga daging sapi lokal mencapai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. menurut yusuf, salah satu pedagang daging sapi di pasar jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.	jakarta - harga daging sapi lokal dan impor di pasar jatinegara alami kenaikan. catat harga daging sapi lokal capai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. turut yusuf, salah satu dagang daging sapi di pasar jatinegara, ia hanya jual 20 kg daging sapi per hari hitung sejak hari ini karena naik harga.
	Filtering	jakarta - harga daging sapi lokal dan impor di pasar jatinegara alami kenaikan. catat harga daging sapi lokal capai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. turut yusuf, salah satu dagang daging sapi di pasar jatinegara, ia hanya jual 20 kg daging sapi per hari hitung sejak hari ini karena naik harga.	jakarta - harga daging sapi lokal impor pasar jatinegara alami kenaikan. catat harga daging sapi lokal capai rp 130.000/kg. lalu, daging sapi beku impor india rp 100.000/kg. turut yusuf, salah satu dagang daging sapi pasar jatinegara, hanya jual 20 kg daging sapi per hari hitung sejak hari karena naik harga.
	Tokenize Kalimat	jakarta - harga daging sapi lokal impor pasar jatinegara mengalami kenaikan. tercatat harga daging sapi lokal mencapai rp 130.000/kg. lalu, daging sapi beku impor india rp 100.000/kg. yusuf, salah pedagang daging sapi pasar jatinegara, menjual 20 kg daging sapi harinya terhitung kenaikan harga.	['jakarta - harga daging sapi lokal impor pasar jatinegara alami kenaikan.', 'catat harga daging sapi lokal capai rp 130.000/kg.', 'lalu, daging sapi beku impor india rp 100.000/kg.', 'turut yusuf, salah satu dagang daging sapi pasar jatinegara, hanya jual 20 kg daging sapi per hari hitung sejak hari karena naik harga.']
	Tokenize Kata	['jakarta - harga daging sapi lokal impor pasar jatinegara alami kenaikan.', 'catat harga daging sapi lokal capai rp 130.000/kg.', 'lalu, daging sapi beku impor india rp 100.000/kg.', 'turut yusuf, salah	['catat', 'harga', 'daging', 'sapi', 'lokal', 'capai', 'rp', '130.000/kg', '.'], ['lalu', ',', 'daging', 'sapi', 'beku', 'impor', 'india', 'rp', '100.000/kg', '.'], ['turut',

No	Fungsi	Input	Hasil yang diharapkan
		satu dagang daging sapi pasar jatinegara, hanya jual 20 kg daging sapi per hari hitung sejak hari karena naik harga.']	'yusuf', ',', 'salah', 'satu', 'dagang', 'daging', 'sapi', 'pasar', 'jatinegara', ',', 'hanya', 'jual', '20', 'kg', 'daging', 'sapi', 'per', 'hari', 'hitung', 'sejak', 'hari', 'karena', 'naik', 'harga', '.']
	Jaro-Winkler Komoditas dan harga	['catat', 'harga', 'daging', 'sapi', 'lokal', 'capai', 'rp', '130.000/kg', '.'], ['lalu', ',', 'daging', 'sapi', 'beku', 'impor', 'india', 'rp', '100.000/kg', '.'], ['turut', 'yusuf', ',', 'salah', 'satu', 'dagang', 'daging', 'sapi', 'pasar', 'jatinegara', ',', 'hanya', 'jual', '20', 'kg', 'daging', 'sapi', 'per', 'hari', 'hitung', 'sejak', 'hari', 'karena', 'naik', 'harga', '.']	Daging sapi, 115000
5	Import data pembanding	Menekan tombol <i>add secondary data</i>	Aplikasi dapat meng-import data excel kedalam database <i>secondary</i>
6	Melihat data	Menekan tombol apply	Aplikasi dapat menampilkan data dalam diagram garis
7	Melihat data dengan data pembanding	Mencentang view secondary data, dan menekan tombol apply	Aplikasi dapat menampilkan data dari sumber primer dan pembanding dalam diagram garis

B. Skenario Uji Akurasi

Untuk mengetahui akurasi dari aplikasi disusun rangkaian pengujian yang terbagi menjadi pengujian akurasi seleksi harga komoditas, pengujian akurasi jaro-winkler, dan pengujian selisih data harga .

B.1. Uji Akurasi Seleksi Harga Komoditas.

Uji akurasi seleksi data harga akan dibagi menjadi uji akurasi dari sumber twitter dan media berita online. Adapun harga komoditas yang dicari pada skenario ini adalah harga komoditas daging sapi dan telur ayam ras. Berikut ini merupakan skenario uji akurasi sumber twitter.

Tabel 3.3 Skenario Uji Akurasi Sumber Twitter

No	Data Tweet	Hasil yang diharapkan
1.	'nya untuk Wilayah Kec.warungkondang\nDi Jual Daging Sapi untuk d antar H-1 Lebaran /Kg Rp.120.000\n*Bonus 1 Liter Minyak Goreng (Berlaku kelipatan).\nMinat Inbo'	Harga diambil dan ditampilkan
2	b'Harga daging sapi meroket sampai Rp. 150.000/kg pun sangat menyentuh... \nHiksss... \xf0\x9f\x98\xad https://t.co/KKOioxNyeN	Harga diambil dan ditampilkan
3	b'Harga Daging Sapi Bertahan Rp 140.000 per Kilogram di Pasar Baru Bekasi - Warta Kota https://t.co/jzqwKHhhuJ	Harga diambil dan ditampilkan
4	b'Hai, Family Midi!\nHari ini kamu bisa TEBUS MURAH INDOFOOD Kecap Manis hanya Rp 5.000! Cukup dengan membeli daging sapi atau ayam menggunakan kartu Ponta di Alfamidi. Jangan sampai terlewatkan ya!\n#AyokeAlfamidi https://t.co/r7kO6M0gJa	Harga tidak diambil dan tidak ditampilkan
5	b'\xf0\x9f\x8f\xa1 SOTO SUMUR\n\xfb\x9f\x94\xb0 DEPAN KELURAHAN SETABELAN BANJARSARI\n\xe2\x8f\xb0 BUKA JAM 06.00-13.00 WIB\n.\n\xfb\x9f\x92\xb0 SOTO DAGING SAPI = Rp. 11.000,-\n\n#Kulinersolo https://t.co/rt4xGBXZIy	Harga tidak diambil dan tidak ditampilkan
6	b'Tahu bakso khas Malang, asli daging sapi\xfb\x9f\x8d\x96 Rp 15.000,-/ pack \nReady di Lovren Gallery Cepu, sebelah barat SMP Negeri 4 Cepu. Buka pukul 13.00-18.00 WIB\n\nOrder by WA, chat saya di 082140409344 \nTerimakasih \xf0\x9f\x98\x8d'	Harga tidak diambil dan tidak ditampilkan
7	b'Hai, Family Midi!\nHari ini kamu bisa TEBUS MURAH INDOFOOD Kecap Manis hanya Rp 5.000! Cukup dengan membeli daging sapi atau ayam menggunakan kartu Ponta di Alfamidi. Jangan sampai terlewatkan ya!\n#AyokeAlfamidi https://t.co/r7kO6M0gJa	Harga tidak diambil dan tidak ditampilkan

No	Data Tweet	Hasil yang diharapkan
8	b'rga daging sapi di Aceh melonjak drastis pada hari meugang atau dua hari jelang Ramadan. Di beberapa wilayah, harganya mencapai Rp 200.000/kg. https://t.co/ACDpNqo0Y'b'rga daging sapi di Aceh melonjak drastis pada hari meugang atau dua hari jelang Ramadan. Di beberapa wilayah, harganya mencapai Rp 200.000/kg. https://t.co/ACDpNqo0Y'	Harga diambil dan ditampilkan
9	b'Harga Telur Ayam Ras Rp 22.000 per Kilo di Kota Banjarmasin, Stok Kemi Normal Pascalebaran https://t.co/u3Yb7dpYRJ'	Harga diambil dan ditampilkan
10	b'Harga Sembako Di Pasar Lakessi Kota Parepare 18 Juni 2019, Telur Ayam Ras Mengalami Penurunan Harga Dari Rp. 23.000/Kg Menjadi Rp. 22.000/Kgnya Sementara Cabai Merah Mengalami Kenaikan Harga Dari Rp. 15.000/Kg\xe2\x80\xa6 https://t.co/EiV3RMm5km'	Harga tidak diambil dan tidak ditampilkan

Selanjutnya adalah skenario pengujian akurasi penyeleksian harga komoditas dari sumber Media Berita online. Adapun data yang digunakan dalam skenario ini diambil dari sumber Liputan 6 dan Detik.com karena proses seleksi menggunakan cara yang sama hanya saja diambil dari sumber yang berbeda. Berikut ini merupakan skenario pengujian akurasi seleksi harga komoditas dari web.

Tabel 3.4 Skenario Uji Akurasi Sumber Media Berita Online

No	Data Berita	Hasil yang Diharapkan
1	Amran mengungkapkan daging sapi yang akan diimpor memiliki kualitas medium dengan rata-rata harga US\$ 3 per kilogram (kg) atau setara dengan Rp 42.600 (kurs Rp 14.200)	Harga diambil dan ditampilkan
2	daging sapi beku Rp 78.00 per kilogram (kg), daging kerbau Rp 60.000 per 900 gram, daging ayam Rp 30.000 per kg	Harga tidak diambil dan tidak ditampilkan

No	Data Berita	Hasil yang Diharapkan
3	Harga ini sudah mengalami kenaikan Rp 1.000/kg sejak sebelum Ramadhan. Tetapi harga ini masih di bawah HAP telur ayam ras Rp 23.000/kg.	Harga tidak diambil dan tidak ditampilkan
4	., "Harga daging standar, Rp 125.000 per kg. Biasa baru naik nanti pas Lebaran," ucap dia.,	Harga diambil dan ditampilkan
5	Jakarta - Harga daging sapi lokal di Pasar Jatinegara mencapai Rp 120.000/kg	Harga diambil dan ditampilkan
6	Pedagang telur lain di Toko Adam juga menjual telur ayam negeri di harga Rp 24 ribu per kg dan telur ayam kampung Rp 2.500 per kg	Harga diambil dan ditampilkan
7	Tercatat harga daging sapi lokal mencapai Rp 130.000/kg.	Harga diambil dan ditampilkan
8	Adapun harga daging sapi Rp 120.000/kg dan daging ayam ras Rp 28.000/kg ekor hidup,	Harga diambil dan ditampilkan
9	Harga daging sapi lokal stabil saja Rp 120.000/kg dari sebelum bulan puasa.	Harga diambil dan ditampilkan
10	Adapun, mengacu pada info pangan DKI Jakarta hari ini maka rerata harga jual telur ayam ras di pasar tradisional saat ini ialah sebesar Rp 24.462 per kg.	Harga diambil dan ditampilkan

B.2 Uji Akurasi Jaro-Winkler

Berikut merupakan skenario pengujian modul jaro winkler untuk mengetahui tingkat akurasi dari modul tersebut.

Tabel 3.5 Skenario Akurasi Jaro Winkler

No	String 1	String 2	Similiarity	Hasil yang diharapkan
1	Dagang	Daging	0,92	Terseleksi
2	Dagi	Daging	0,93	Terseleksi
3	Sapi	Saki	0,87	Terseleksi
4	Sapi	Pisa	0,83	Tidak Terseleksi
5	Telur	Tekor	0,79	Tidak Terseleksi
6	Telur	Telor	0,91	Terseleksi
7	Ayam	Maya	0,83	Tidak Terseleksi
8	Ayam	Ayan	0,93	Terseleksi
9	Ayam	Ayaam	0,95	Terseleksi
10	Daging	Daagiing	0,93	Terseleksi

3.5. Deployment

Dikarenakan tujuan dari penelitian ini adalah untuk menyelesaikan tugas akhir sebagai salah satu syarat kelulusan maka penelitian ini tidak melalui proses rilis aplikasi



BAB IV

HASIL DAN PEMBAHASAN

Bab ini akan menjelaskan hasil dari perancangan desain sistem atau aplikasi yang telah digambarkan sebelumnya. Selain itu tahap ini juga akan memaparkan perangkat lunak dan perangkat keras yang digunakan dalam implementasi dari perancangan desain sistem atau aplikasi.

Berikut ini merupakan perangkat lunak yang digunakan untuk mengimplementasikan sistem atau aplikasi :

1. Sistem operasi yang digunakan adalah Windows 10 Home
2. *Database management system* yang digunakan adalah SQL Server 2017
3. Bahasa pemrograman yang digunakan adalah Python 3.6
4. *Package distributor* yang digunakan adalah Anaconda Navigator 1.9.7
5. *Integrated development environment* (IDE) yang digunakan adalah Spyder

3.3.4

Dibawah ini adalah spesifikasi perangkat keras yang digunakan dalam implementasi sistem atau aplikasi:

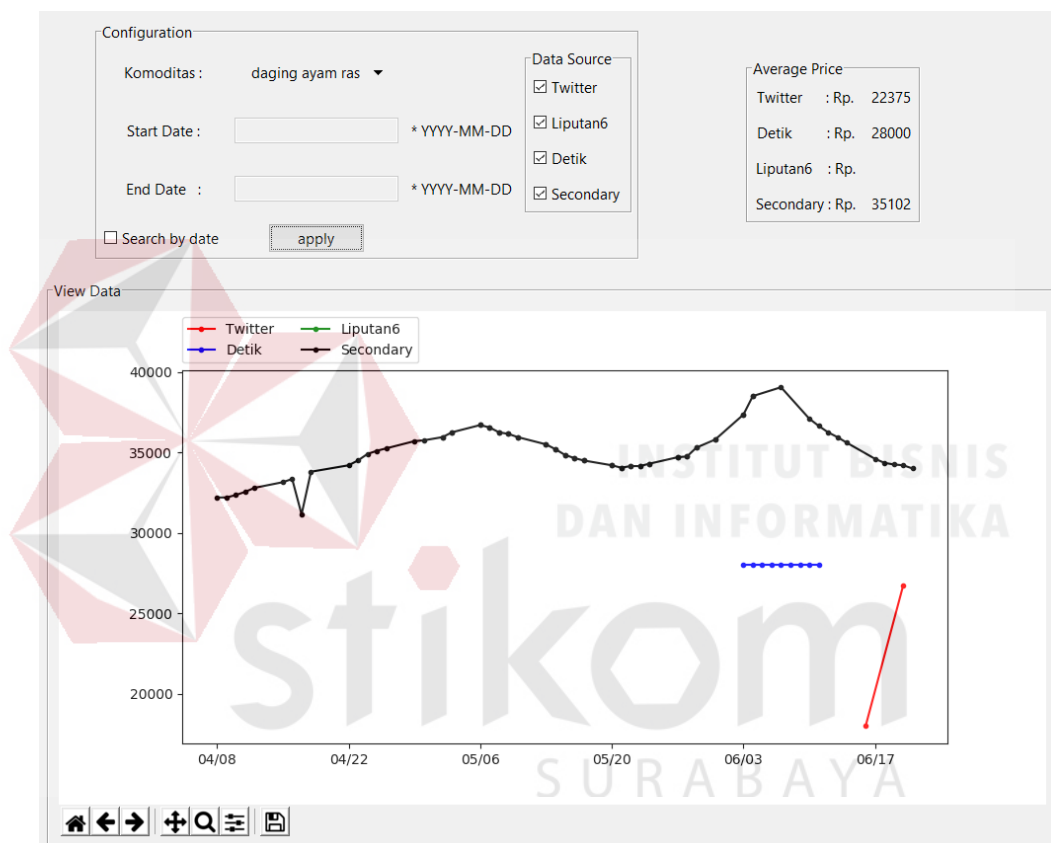
1. Intel Core i7 7700HQ up to 3.8 ghz
2. 16 gigabyte RAM
3. Ruang Penyimpanan lebih dari 500 Mb
4. Monitor dengan resolusi 1080p

4.1 Implementasi

Dalam tahap ini akan terbagi lagi menjadi dua bagian yaitu implementasi sistem serta implementasi *user interface*(UI).

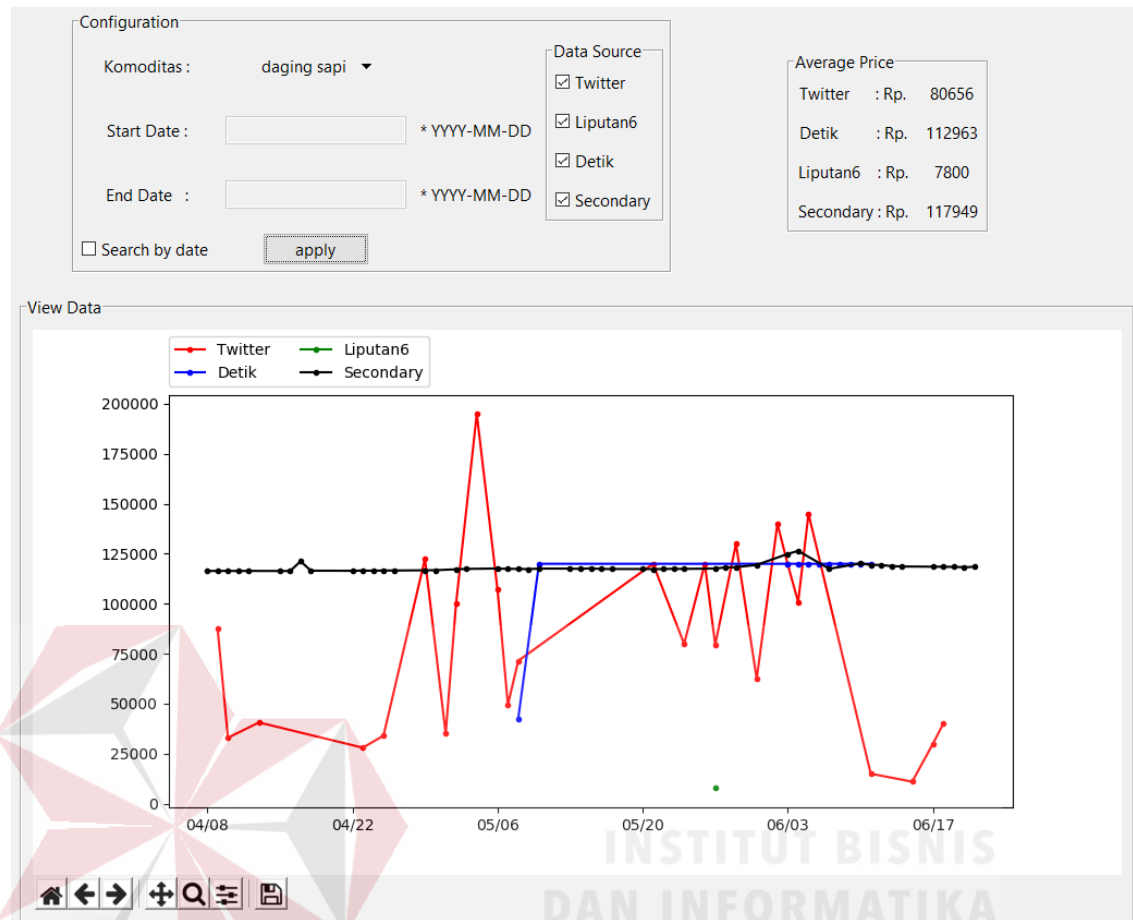
4.1.1 Implementasi Sistem

Dalam sumber data pembanding, tidak semua komoditas tersedia datanya. Adapun data yang disediakan antara lain adalah daging ayam ras, daging sapi, dan telur ayam ras. Dibawah ini adalah perbandingan antara data daging ayam ras yang didapat dari data pembanding dan data yang didapat dari aplikasi persumbernya.



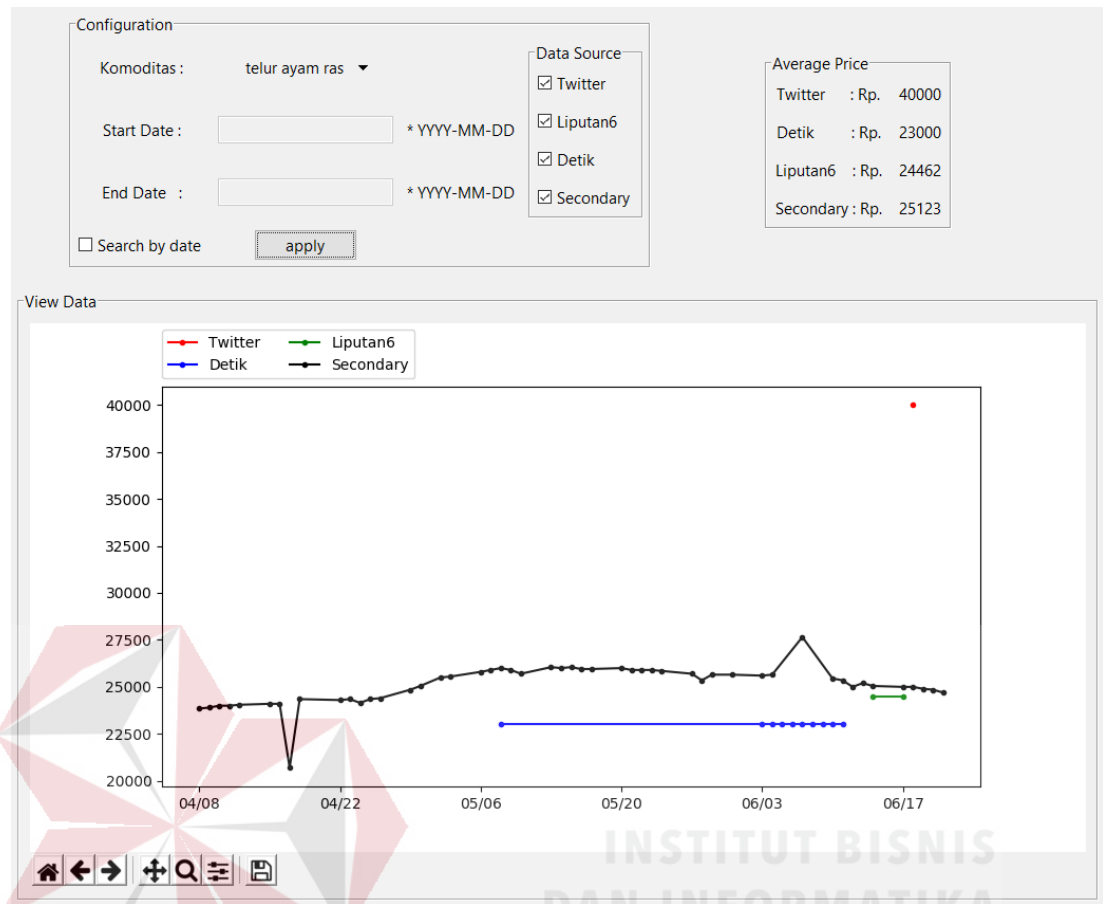
Gambar 4.1 Visualisasi Harga Daging Ayam Ras

Selanjutnya adalah perbandingan antara data daging sapi yang didapat dari data pembanding dan data yang didapat dari aplikasi persumbernya.



Gambar 4.2 Visualisasi Harga Daging Sapi

Terakhir merupakan perbandingan antara data telur ayam ras yang didapat dari data pembanding dan data yang didapat dari aplikasi persumbernya.



Gambar 4.3 Visualisasi Harga Telur Ayam Ras

Untuk data harga yang didapat dari aplikasi merupakan harga rata-rata sehingga kecil kemungkinannya untuk sama persis dengan harga pembanding. Seperti yang dapat dilihat pada gambar diatas untuk perbandingan harga daging sapi, twitter memang mempunyai data lebih banyak tapi harga yang terseleksi lebih tidak akurat karena banyaknya fluktuasi yang terlalu tajam dibandingkan data dari media berita online. Hal ini didukung oleh penulisan media sosial yang lebih tidak teratur sehingga memerlukan kamus pengecekan “cek_tweet” pada database yang lebih luas untuk meningkatkan keakuratan seleksi. “cek_tweet” digunakan untuk menampung kata-kata yang membuat sebuah tweet dianggap tidak relevan. Contoh adalah tweet pada jenis komoditas jeruk yang berbunyi “Dijual Jus Jeruk Rp.1000”.

Kata “Jus” akan dimasukkan kedalam database “cek_tweet” sehingga tweet tersebut secara otomatis dianggap tidak relevan. Selain itu juga terdapat beberapa fluktuasi yang tajam pada harga komoditas pangan seperti harga daging sapi pada liputan 6 dan Detik. Adapun beberapa kemungkinan yang menyebabkan hal ini adalah adanya beberapa tag iklan milik google yang terdapat didalam text berita yang dapat mengandung harga yang masih terambil saat proses scrapping. Memang tag tersebut akan hilang saat proses *preprocessing* namun text didalam tag akan tetap ada dan dianggap sebagai text berita. Selain itu juga algoritma juga tidak mengecek text sebelum dan sesudah nama komoditas, sehingga jika semisal terdapat kata-kata seperti “Bakso daging sapi” aplikasi tetap menganggap kata tersebut termasuk komoditas daging sapi. Algoritma juga tidak memeriksa harga persatuan yang dimiliki komoditas apakah per kilo maupun per ons. Namun data-data tersebut jarang sekali ditemui pada data berita media online.

Memang Data yang didapat dari media berita online meskipun lebih stabil tetapi tetap saja lebih sedikit dari media sosial online, hal ini juga dipengaruhi dekatnya waktu pemilu saat pengambilan data sehingga banyak berita yang berisi ekonomi politik dibandingkan berita yang mengandung data harga komoditas.

Hal tersebut juga berlaku kepada perbandingan komoditas lainnya. Rata-rata twitter memiliki lebih banyak data tetapi media berita online masih terlihat memiliki fluktuasi yang lebih normal. Tetapi ada satu komoditas dimana Detik dan liputan 6 memiliki lebih banyak data, yaitu komoditas telur ayam ras. Pada komoditas ini, twitter hanya memiliki satu data saja. Selain itu untuk komoditas daging ayam ras tidak ditemukan pada sumber data Liputan6.

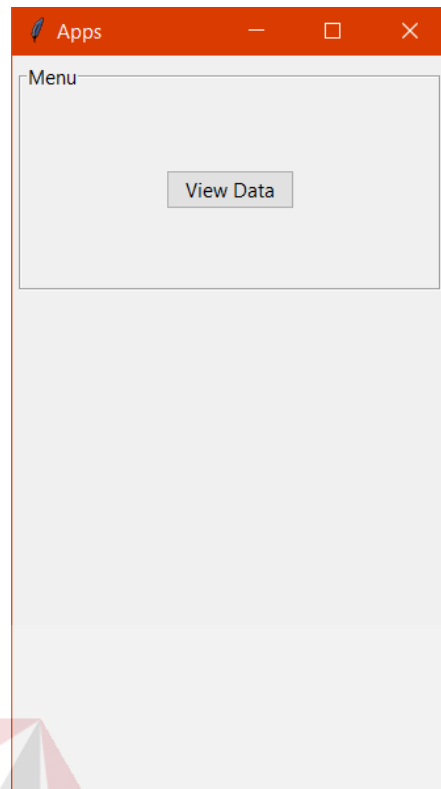
4.1.2 Implementasi *User Interface*(UI)

Pada implementasi UI terdapat dua perspektif yaitu dari sudut pandang pengguna dan sudut pandang admin. Tetapi terdapat implementasi UI yang sama yaitu pada UI *popup* untuk melihat harga. Berikut ini adalah implementasi dari tampilan *Login* :



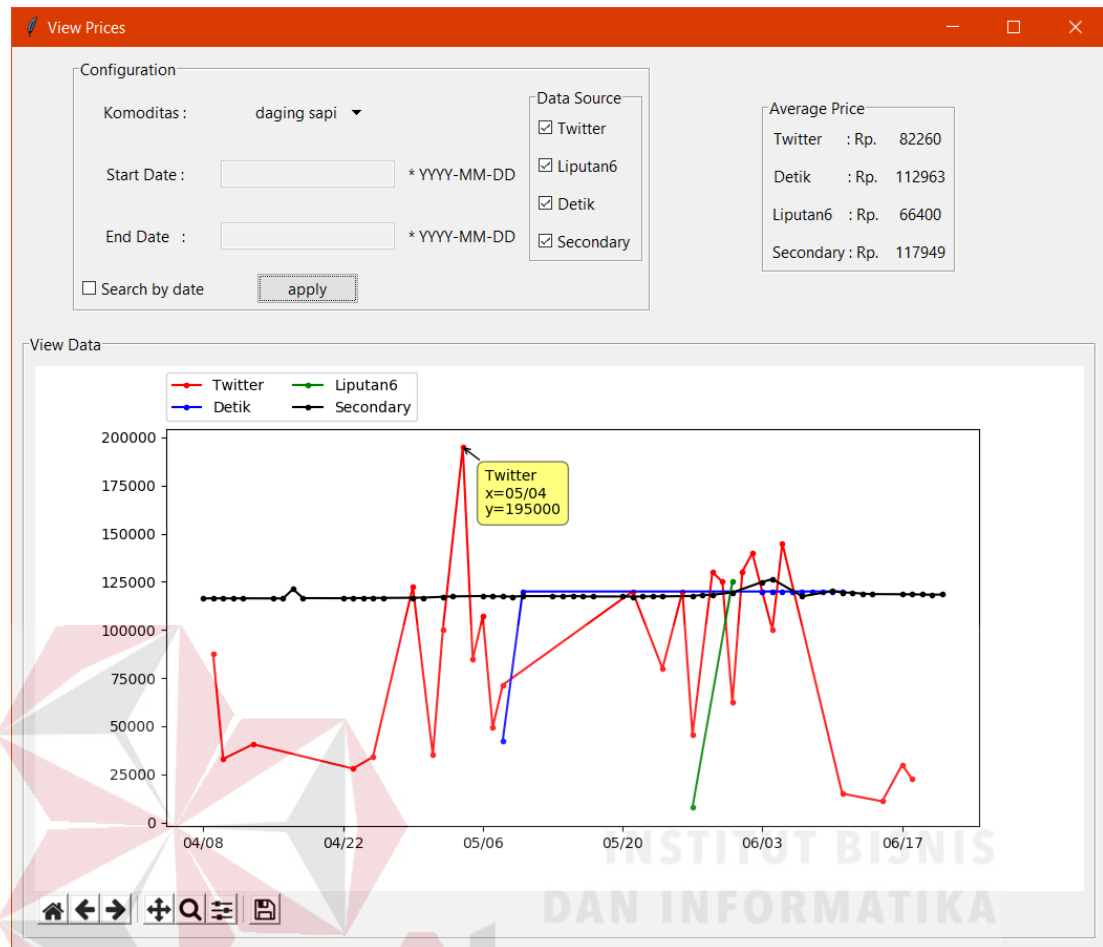
Gambar 4.4 Implementasi Halaman Login

Pada halaman ini pengguna mengisi *username* dan *password* dan menekan tombol *login*. Halaman selanjutnya akan bergantung kepada role dari pengguna apabila pengguna adalah pengguna biasa(*user*), maka aplikasi akan menampilkan halaman utama pengguna, jika role pengguna adalah admin, maka aplikasi akan menampilkan halaman utama admin. Dibawah ini merupakan gambar dari implementasi halaman utama pengguna.



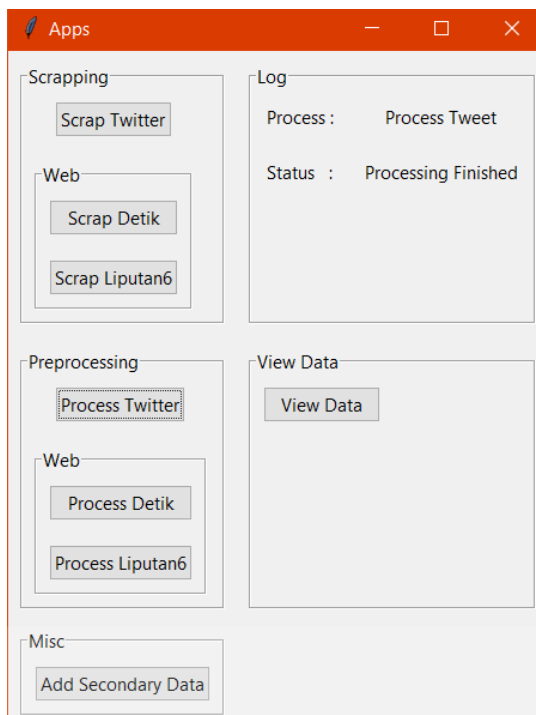
Gambar 4.5 Halaman Pengguna.

Pada halaman pengguna ini terdapat tombol yang akan memunculkan *popup* yang berguna untuk melihat harga komoditas. Adapun implementasi dari halaman *popup* itu sendiri dapat dilihat pada gambar 4.6. Contoh gambar tersebut memperlihatkan data dengan komoditas daging sapi dari semua sumber dan melakukan klik pada diagram garis pada tanggal 2019-05-04. Halaman ini juga akan diakses oleh admin dari halaman utama admin. Disini pengguna atau admin dapat memilih komoditas yang ingin dilihat. Pengguna / admin juga dapat mencentang *search by date* untuk mengaktifkan *entry start date* dan *end date* untuk memasukkan tanggal yang diinginkan. Selain itu detail harga pada diagram dapat dilihat dengan melakukan klik pada lingkaran yang ada di diagram



Gambar 4.6 Popup Lihat Harga

Didalam label frame *View Data* terdapat kanvas tempat diagram garis digambarkan nantinya. Selain itu dibawah kanvas terdapat *navigation bar* yang berfungsi untuk membantu menavigasi diagram garis. Adapun penjelasan ikon dari kanan ke kiri adalah; ikon untuk menyimpan diagram garis sebagai gambar, ikon untuk mengatur lebar dan tinggi diagram, ikon untuk *zoom*, ikon untuk menavigasi menggunakan mouse, ikon *redo* ke tampilan sebelumnya, ikon *undo* ke tampilan sebelumnya, tombol kembali ke tampilan awal.



Gambar 4.7 Tampilan Admin

Gambar diatas adalah tampilan yang ditampilkan untuk admin. Pada label *frame Scrapping* adalah beberapa tombol untuk mengambil data dari sumber yang tertulis pada masing masing tombol. Untuk label *frame Preprocessing* berisi beberapa tombol untuk memproses data mentah yang diambil sebelumnya, juga dibagi sesuai sumbernya. Lalu pada label *frame Misc* berisi tombol untuk meng-*import* data pembandingan. Pada label *frame Log* terdapat rangkaian label yang akan memberitahu pengguna proses yang sedang berlangsung dan status dari proses tersebut. Semua proses didalam button pada halaman ini kecuali button view data, akan ditampilkan pada label *frame Log*. Jika proses dimulai akan muncul tulisan “[nama proses],please wait” pada status, jika selesai tulisan itu akan menjadi “[nama proses],process completed”. Terakhir adalah label *frame view data* yang akan memunculkan *popup* untuk melihat harga komoditas.

4.2 Uji Coba

Uji coba aplikasi akan dibagi menjadi dua bagian yaitu:

4.2.1 Uji Coba Fungsi

Berikut ini merupakan hasil dari implementasi skenario testing yang telah disusun pada bab sebelumnya. Untuk lebih detilnya dapat dilihat pada tabel dibawah ini

Tabel 4.2 Implementasi Testing Fungsi

No	Fungsi	Input	Hasil yang diharapkan	Hasil Implementasi
1	<i>Login Pengguna</i>	User,user	Aplikasi menampilkan halaman pengguna	Aplikasi berhasil menampilkan halaman pengguna
2	<i>Login Admin</i>	Admin,admin	Aplikasi menampilkan halaman admin	Aplikasi berhasil menampilkan halaman admin
3	<i>Scrapping</i>	Menekan tombol <i>scrap</i> twitter	Aplikasi dapat mengambil data dari sumber dan menyimpan didalam <i>database</i> tweetR	Data berhasil diambil dari sumber dan disimpan di <i>database</i> tweetR
		Menekan tombol <i>scrap</i> detik	Aplikasi dapat mengambil data dari sumber dan menyimpan didalam <i>database</i> detikR	Data berhasil diambil dari sumber dan disimpan di <i>database</i> detikR
		Menekan tombol <i>scrap</i> liputan 6	Aplikasi dapat mengambil data dari sumber dan menyimpan didalam <i>database</i> lip6R	Data berhasil diambil dari sumber dan disimpan di <i>database</i> lip6R

No	Fungsi	Input	Hasil yang diharapkan	Hasil Implementasi
4	<i>Processing</i>	Menekan tombol <i>process</i> twitter	Aplikasi dapat melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> tweetF	Aplikasi berhasil melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> tweetF
		Menekan tombol <i>process</i> detik	Aplikasi dapat melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> detikF.	Aplikasi berhasil melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> detikF.
		Menekan tombol <i>process</i> liputan6	Aplikasi dapat melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> lip6F.	Aplikasi berhasil melakukan proses <i>preprocessing</i> dan seleksi jaro winkler terhadap data mentah dan menyimpan data tersebut di <i>database</i> lip6F.
	Membersihkan Tag	Jakarta - Harga daging sapi lokal dan impor di Pasar Jatinegara mengalami kenaikan. Tercatat harga daging sapi lokal mencapai Rp 130.000/kg. Lalu, daging sapi beku impor dari India Rp	Jakarta - Harga daging sapi lokal dan impor di Pasar Jatinegara mengalami kenaikan. Tercatat harga daging sapi lokal mencapai Rp 130.000/kg. Lalu, daging sapi beku impor dari India Rp	Jakarta - Harga daging sapi lokal dan impor di Pasar Jatinegara mengalami kenaikan. Tercatat harga daging sapi lokal mencapai Rp 130.000/kg. Lalu, daging sapi beku impor dari India Rp

No	Fungsi	Input	Hasil yang diharapkan	Hasil Implementasi
		100.000/kg. Menurut Yusuf, salah satu pedagang daging sapi di Pasar Jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga. googletag.cmd.push	100.000/kg. Menurut Yusuf, salah satu pedagang daging sapi di Pasar Jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.	100.000/kg. Menurut Yusuf, salah satu pedagang daging sapi di Pasar Jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.
	Case Folding	Jakarta - Harga daging sapi lokal dan impor di Pasar Jatinegara mengalami kenaikan. Tercatat harga daging sapi lokal mencapai Rp 130.000/kg. Lalu, daging sapi beku impor dari India Rp 100.000/kg. Menurut Yusuf, salah satu pedagang daging sapi di Pasar Jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.	jakarta - harga daging sapi lokal dan impor di pasar jatinegara mengalami kenaikan. tercatat harga daging sapi lokal mencapai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. menurut yusuf, salah satu pedagang daging sapi di pasar jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.	jakarta - harga daging sapi lokal dan impor di pasar jatinegara mengalami kenaikan. tercatat harga daging sapi lokal mencapai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. menurut yusuf, salah satu pedagang daging sapi di pasar jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.

No	Fungsi	Input	Hasil yang diharapkan	Hasil Implementasi
	Stemming	jakarta - harga daging sapi lokal dan impor di pasar jatinegara mengalami kenaikan. tercatat harga daging sapi lokal mencapai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. menurut yusuf, salah satu pedagang daging sapi di pasar jatinegara, ia hanya menjual 20 kg daging sapi per harinya terhitung sejak hari ini karena kenaikan harga.	jakarta - harga daging sapi lokal dan impor di pasar jatinegara alami kenaikan. catat harga daging sapi lokal capai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. turut yusuf, salah satu dagang daging sapi di pasar jatinegara, ia hanya jual 20 kg daging sapi per hari hitung sejak hari ini karena naik harga.	jakarta - harga daging sapi lokal dan impor di pasar jatinegara alami kenaikan. catat harga daging sapi lokal capai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. turut yusuf, salah satu dagang daging sapi di pasar jatinegara, ia hanya jual 20 kg daging sapi per hari hitung sejak hari ini karena naik harga.
	Filtering	jakarta - harga daging sapi lokal dan impor di pasar jatinegara alami kenaikan. catat harga daging sapi lokal capai rp 130.000/kg. lalu, daging sapi beku impor dari india rp 100.000/kg. turut yusuf, salah satu dagang daging sapi di pasar jatinegara, ia hanya jual 20 kg daging sapi per hari hitung sejak hari ini karena naik harga.	jakarta - harga daging sapi lokal impor pasar jatinegara alami kenaikan. catat harga daging sapi lokal capai rp 130.000/kg. lalu, daging sapi beku impor india rp 100.000/kg. turut yusuf, salah satu dagang daging sapi pasar jatinegara, hanya jual 20 kg daging sapi per hari hitung sejak hari ini karena naik harga.	jakarta - harga daging sapi lokal impor pasar jatinegara alami kenaikan. catat harga daging sapi lokal capai rp 130.000/kg. lalu, daging sapi beku impor india rp 100.000/kg. turut yusuf, salah satu dagang daging sapi pasar jatinegara, hanya jual 20 kg daging sapi per hari hitung sejak hari ini karena naik harga.

No	Fungsi	Input	Hasil yang diharapkan	Hasil Implementasi
	Tokenize Kalimat	jakarta - harga daging sapi lokal impor pasar jatinegara mengalami kenaikan. tercatat harga daging sapi lokal mencapai rp 130.000/kg. lalu, daging sapi beku impor india rp 100.000/kg. yusuf, salah satu pedagang daging sapi pasar jatinegara, menjual 20 kg daging sapi harinya terhitung kenaikan harga.	['jakarta - harga daging sapi lokal impor pasar jatinegara alami kenaikan.', 'catat harga daging sapi lokal capai rp 130.000/kg.', 'lalu, daging sapi beku impor india rp 100.000/kg.', 'turut yusuf, salah satu dagang daging sapi pasar jatinegara, hanya jual 20 kg daging sapi per hari hitung sejak hari karena naik harga.']	['jakarta - harga daging sapi lokal impor pasar jatinegara alami kenaikan.', 'catat harga daging sapi lokal capai rp 130.000/kg.', 'lalu, daging sapi beku impor india rp 100.000/kg.', 'turut yusuf, salah satu dagang daging sapi pasar jatinegara, hanya jual 20 kg daging sapi per hari hitung sejak hari karena naik harga.']
	Tokenize Kata	['jakarta - harga daging sapi lokal impor pasar jatinegara alami kenaikan.', 'catat harga daging sapi lokal capai rp 130.000/kg.', 'lalu, daging sapi beku impor india rp 100.000/kg.', 'turut yusuf, salah satu dagang daging sapi pasar jatinegara, hanya jual 20 kg daging sapi per hari hitung sejak hari karena naik harga.']	['catat', 'harga', 'daging', 'sapi', 'lokal', 'capai', 'rp', '130.000/kg', '.'], ['lalu', ',', 'daging', 'sapi', 'beku', 'impor', 'india', 'rp', '100.000/kg', '.'], ['turut', 'yusuf', ',', 'salah', 'satu', 'dagang', 'daging', 'sapi', 'pasar', 'jatinegara', ',', 'hanya', 'jual', '20', 'kg', 'daging', 'sapi', 'per', 'hari', 'hitung', 'sejak', 'hari', 'karena', 'naik', 'harga', '.']	['catat', 'harga', 'daging', 'sapi', 'lokal', 'capai', 'rp', '130.000/kg', '.'], ['lalu', ',', 'daging', 'sapi', 'beku', 'impor', 'india', 'rp', '100.000/kg', '.'], ['turut', 'yusuf', ',', 'salah', 'satu', 'dagang', 'daging', 'sapi', 'pasar', 'jatinegara', ',', 'hanya', 'jual', '20', 'kg', 'daging', 'sapi', 'per', 'hari', 'hitung', 'sejak', 'hari', 'karena', 'naik', 'harga', '.']

No	Fungsi	Input	Hasil yang diharapkan	Hasil Implementasi
	Jaro-Winkler Komoditas dan harga	['catat', 'harga', 'daging', 'sapi', 'lokal', 'capai', 'rp', '130.000/kg', '.'], ['lalu', ',', 'daging', 'sapi', 'beku', 'impor', 'india', 'rp', '100.000/kg', '.'], ['turut', 'yusuf', ',', 'salah', 'satu', 'dagang', 'daging', 'sapi', 'pasar', 'jatinegara', ',', 'hanya', 'jual', '20', 'kg', 'daging', 'sapi', 'per', 'hari', 'hitung', 'sejak', 'hari', 'karena', 'naik', 'harga', '.']	Daging sapi, 115000	Daging sapi, 115000
5	Import data pembandingan	Menekan tombol <i>add secondary data</i>	Aplikasi dapat meng-import data excel kedalam database <i>secondary</i>	Aplikasi berhasil meng-import data excel kedalam database <i>secondary</i>
6	Melihat data	Menekan tombol apply	Aplikasi dapat menampilkan data dalam diagram garis	Aplikasi berhasil menampilkan data dalam diagram garis
7	Melihat data dengan data pembandingan	Mencentang view <i>secondary data</i> , dan menekan tombol apply	Aplikasi dapat menampilkan data dari sumber primer dan pembandingan dalam diagram garis	Aplikasi berhasil menampilkan data dari sumber primer dan pembandingan dalam diagram garis

4.2.2 Uji Akurasi

Uji akurasi akan dibagi menjadi pengujian akurasi seleksi harga komoditas, pengujian akurasi jaro-winkler, dan pengujian selisih data harga.

A. Uji Akurasi Seleksi harga komoditas

Uji akurasi akan dibagi menjadi uji akurasi dari sumber Twitter dan media berita online. Berikut merupakan cara pengujian akurasi pada sumber Twitter.

Tabel 4.3 Uji Akurasi Sumber Twitter

No	Data Tweet	Hasil yang diharapkan	Hasil Implementasi	Kesimpulan
1.	'nya untuk Wilayah Kec.warungkondang\nDi Jual Daging Sapi untuk d antar H-1 Lebaran /Kg Rp.120.000\n*Bonus 1 Liter Minyak Goreng (Berlaku kelipatan).\nMinat Inbo'	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
2	b'Harga daging sapi meroket sampai Rp. 150.000/kg pun sangat menyentuh... \nHiksss... \xf0\x9f\x98\xad https://t.co/KKOioxNyeN'	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
3	b'Harga Daging Sapi Bertahan Rp 140.000 per Kilogram di Pasar Baru Bekasi - Warta Kota https://t.co/jzqwKHhhuJ'	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
4	b'Hai, Family Midi!\nHari ini kamu bisa TEBUS MURAH INDOFOOD Kecap Manis hanya Rp 5.000! Cukup dengan membeli daging sapi atau ayam menggunakan kartu Ponta di Alfamidi. Jangan sampai terlewatkan ya!\n#AyokeAlfamidi https://t.co/r7kO6M0gJa'	Harga tidak diambil dan tidak ditampilkan	Harga tidak diambil dan tidak ditampilkan	Benar

No	Data Tweet	Hasil yang diharapkan	Hasil Implementasi	Kesimpulan
5	b'\xf0\x9f\x8f\xa1 SOTO SUMUR\n\xfb\x9f\x94\xb0 DEPAN KELURAHAN SETABELAN BANJARSARI\n\xe2\x8f\xbf 0 BUKA JAM 06.00-13.00 WIB\n.\n\xfb\x9f\x92\xb0 SOTO DAGING SAPI = Rp. 11.000,-\n\n#Kulinersolo https://t.co/rt4xGBXZIy'	Harga tidak diambil dan tidak ditampilkan	Harga diambil dan ditampilkan	Salah
6	b'Tahu bakso khas Malang, asli daging sapi\xfb\x9f\x8d\x96 Rp 15.000,-/ pack \nReady di Lovren Gallery Cepu, sebelah barat SMP Negeri 4 Cepu. Buka pukul 13.00-18.00 WIB\n\nOrder by WA, chat saya di 082140409344 \nTerimakasih \xf0\x9f\x98\x8d'	Harga tidak diambil dan tidak ditampilkan	Harga diambil dan ditampilkan	Salah
7	b'Hai, Family Midi!\nHari ini kamu bisa TEBUS MURAH INDOFOOD Kecap Manis hanya Rp 5.000! Cukup dengan membeli daging sapi atau ayam menggunakan kartu Ponta di Alfamidi. Jangan sampai terlewatkan ya!\n\n#AyokeAlfamidi https://t.co/r7kO6M0gJa'	Harga tidak diambil dan tidak ditampilkan	Harga tidak diambil dan tidak ditampilkan	Benar
8	b'rga daging sapi di Aceh melonjak drastis pada hari meugang atau dua hari jelang Ramadan. Di beberapa wilayah, harganya mencapai Rp 200.000/kg. https://t.co/ACDpNqo0Y'b'rga daging sapi di Aceh melonjak drastis pada hari meugang atau dua hari jelang Ramadan. Di beberapa wilayah, harganya	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar

No	Data Tweet	Hasil yang diharapkan	Hasil Implementasi	Kesimpulan
	mencapai Rp 200.000/kg. https://t.co/ACDpNqo0Y			
9	b'Harga Telur Ayam Ras Rp 22.000 per Kilo di Kota Banjarmasin, Stok Kembal Normal Pascalebaran https://t.co/u3Yb7dpYRJ	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
10	b'Harga Sembako Di Pasar Lakessi Kota Parepare 18 Juni 2019, Telur Ayam Ras Mengalami Penurunan Harga Dari Rp. 23.000/Kg Menjadi Rp. 22.000/Kgnya Sementara Cabai Merah Mengalami Kenaikan Harga Dari Rp. 15.000/Kg https://t.co/EiV3RMm5km	Harga tidak diambil dan tidak ditampilkan	Harga diambil dan ditampilkan	Salah

Dari tabel uji akurasi Twitter diatas dapat dilihat dari 10 data yang diuji terdapat 3 data yang salah sehingga dari data tersebut dapat dihitung tingkat akurasi penyeleksian harga komoditas dari sumber Twitter adalah 7/10 atau jika dikonversikan ke persen adalah 70%.

Selanjutnya adalah cara pengujian akurasi penyeleksian harga komoditas dari sumber Media Berita online

Tabel 4.4 Uji Akurasi Sumber Media Berita Online

No	Data Berita	Hasil yang Diharapkan	Hasil Implementasi	Kesimpulan
1	Amran mengungkapkan daging sapi yang akan diimpor memiliki kualitas medium dengan rata-rata harga US\$ 3 per kilogram (kg) atau setara dengan Rp 42.600 (kurs Rp 14.200)	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar

No	Data Berita	Hasil yang Diharapkan	Hasil Implementasi	Kesimpulan
2	daging sapi beku Rp 78.00 per kilogram (kg), daging kerbau Rp 60.000 per 900 gram, daging ayam Rp 30.000 per kg	Harga tidak diambil dan tidak ditampilkan	Harga diambil dan ditampilkan	Salah
3	Harga ini sudah mengalami kenaikan Rp 1.000/kg sejak sebelum Ramadhan. Tetapi harga ini masih di bawah HAP telur ayam ras Rp 23.000/kg.	Harga tidak diambil dan tidak ditampilkan	Harga diambil dan ditampilkan	Salah
4	., "Harga daging standar, Rp 125.000 per kg. Biasa baru naik nanti pas Lebaran," ucap dia.,	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
5	Jakarta - Harga daging sapi lokal di Pasar Jatinegara mencapai Rp 120.000/kg	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
6	Pedagang telur lain di Toko Adam juga menjual telur ayam negeri di harga Rp 24 ribu per kg dan telur ayam kampung Rp 2.500 per kg	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
7	Tercatat harga daging sapi lokal mencapai Rp 130.000/kg.	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
8	Adapun harga daging sapi Rp 120.000/kg dan daging ayam ras Rp 28.000/kg ekor hidup,	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
9	Harga daging sapi lokal stabil saja Rp 120.000/kg dari sebelum bulan puasa.	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar
10	Adapun, mengacu pada info pangan DKI Jakarta hari ini maka rerata harga jual telur ayam ras di pasar tradisional saat ini ialah sebesar Rp 24.462 per kg.	Harga diambil dan ditampilkan	Harga diambil dan ditampilkan	Benar

Dari tabel uji akurasi Web. diatas dapat dilihat dari 10 data yang diuji terdapat 2 data yang salah sehingga dari data tersebut dapat dihitung tingkat akurasi penyeleksian harga komoditas dari sumber Twitter adalah 8/10 atau jika

dikonversikan ke persen adalah 80%.Berikut ini adalah contoh perhitungan detail tingkat akurasi serta total rata-rata dari kedua pengujian yang dilakukan

Tabel 4.5 Contoh Perhitungan Akurasi Rata-Rata

No	Sumber	Akurasi
1	Twitter	70%
2	Media Berita Online	80%
Akurasi Rata-Rata Seleksi Harga Komoditas		75%

B. Uji Akurasi Jaro-Winkler

Berikut merupakan hasil pengujian implementasi modul jaro-winkler pada aplikasi.

Tabel 4.6 Akurasi Jaro Winkler

No	String 1	String 2	Similiarity	Hasil yang diharapkan	Hasil Implementasi	Kesimpulan
1	Dagang	Daging	0,92	Terseleksi	Terseleksi	Benar
2	Dagi	Daging	0,93	Terseleksi	Terseleksi	Benar
3	Sapi	Sapik	0,96	Terseleksi	Terseleksi	Benar
4	Sapi	Pisa	0,83	Tidak Terseleksi	Tidak Terseleksi	Benar
5	Telur	Tekor	0,79	Tidak Terseleksi	Tidak Terseleksi	Benar
6	Telur	Telor	0,91	Terseleksi	Terseleksi	Benar
7	Ayam	Maya	0,83	Tidak Terseleksi	Tidak Terseleksi	Benar
8	Ayam	Ayan	0,93	Terseleksi	Terseleksi	Benar
9	Ayam	Ayaam	0,95	Terseleksi	Terseleksi	Benar
10	Daging	Daagiing	0,93	Terseleksi	Terseleksi	Benar

4.3 Evaluasi

Pada batasan masalah disebutkan bahwa aplikasi akan mencari 3 komoditas dengan tingkat konsumsi tertinggi per-kategori pada tahun 2016 yang disebutkan didalam buku Statistik Pertanian. Tetapi pada implementasinya tidak semua komoditas tersebut dapat diambil karena beberapa pertimbangan. Adapun beberapa komoditas tersebut antara lain :

1. Pada kategori sayur sayuran nomor 1 dan 2 tertinggi adalah sayur sop dan sayur asam. Kedua komoditas ini adalah produk jadi yang harganya akan bervariasi dikarenakan perbedaan takaran bumbu yang digunakan dalam proses produksi dan keuntungan yang ditarget oleh penjual
2. Pada kategori kacang kacangan nomor 1 tertinggi adalah tahu. Tetapi tahu memiliki dua makna yaitu tahu(makanan) dan tahu(mengetahui) yang pada penerapannya di scrapping twitter mayoritas data yang terjaring adalah data tahu dengan makna mengetahui.
3. Pada kategori buah-buahan yang tertinggi adalah komoditas pisang lainnya. Tetapi komoditas ini tidak dispesifikkan jenisnya dan proporsinya
4. Pada kategori susu semua komoditas yang memiliki data di tahun 2016 susu prosesan pabrik yang setiap merk dapat memiliki harga yang berbeda dan juga bergantung terhadap banyaknya keuntungan yang ditetapkan penjual
5. Kategori daging lainnya dan lainnya tidak diambil karena banyaknya data yang tidak terjaring dari tahun ke tahun dan tidak spesifik daging apa yang termasuk lainnya serta pembagian persennannya.

Maka dari itu, dalam implementasi aplikasi ini adapun komoditas yang dicari antara lain adalah; bayam, daging ayam ras, daging ayam kampung, daging sapi, jagung, jeruk, kangkung, ketela rambat, rambutan, semangka, ketela pohon, talas, telur ayam ras, telur ayam kampung, telur puyuh, tepung terigu, tomat.

Untuk penerapan algoritma dalam penyeleksian data harga komoditas pangan murni hanya mencari nama komoditas dan harganya saja pada suatu teks. Sehingga faktor faktor lain seperti apakah komoditas tersebut adalah barang impor ataupun jika harga yang ada diteks adalah target penyebaran pemerintah maupun

data dari tahun lalu tidak diperiksa. Selain itu algoritma juga tidak memeriksa satuan dari harga apakah perkilo atau selusin. Algoritma juga akan menghitung rata-rata harga komoditas jika ditemukan ada lebih dari satu harga pada tanggal yang sama. Hal ini dilakukan karena untuk visualisasi data hanya pertanggal saja.

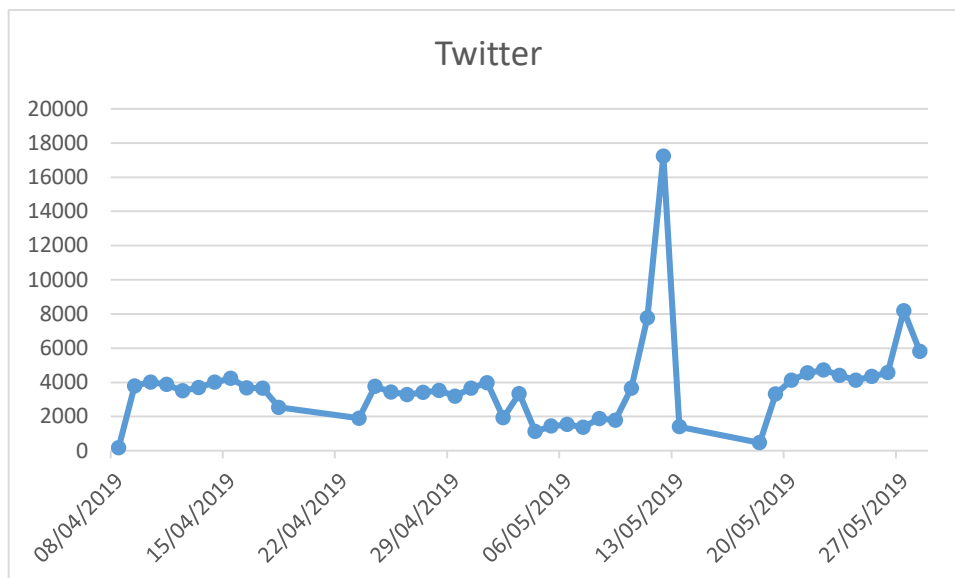
4.3.1. Karakteristik Data Source

Adapun data yang banyak data mentah yang didapat dari process scrapping dari masing masing sumber dan banyak data yang mengandung data harga komoditas antara lain adalah.

Tabel 4.7 Data Yang Didapat Dari *Scrapping*

No	Nama Sumber	Data Mentah
1	Twitter	92204
2	Detik	2903
3	Liputan 6	3900

Dalam Pengambilan data dari Twitter digunakan kata kunci untuk memudahkan pencarian data, pada tanggal 8 April sampai dengan 29 Mei data dicari dengan kata kunci nama komoditas saja. Untuk itu hasil data yang didapat sangat banyak dan banyak yang tidak relevan. Banyaknya data hasil *scrapping* Twitter sebelum tanggal 29 mei dapat dilihat pada gambar 4.8. Karena itulah pada untuk tanggal 30 Mei sampai 23 juni data diambil menggunakan katakunci “[nama komoditas] rp” untuk mendapat data yang lebih relevan.

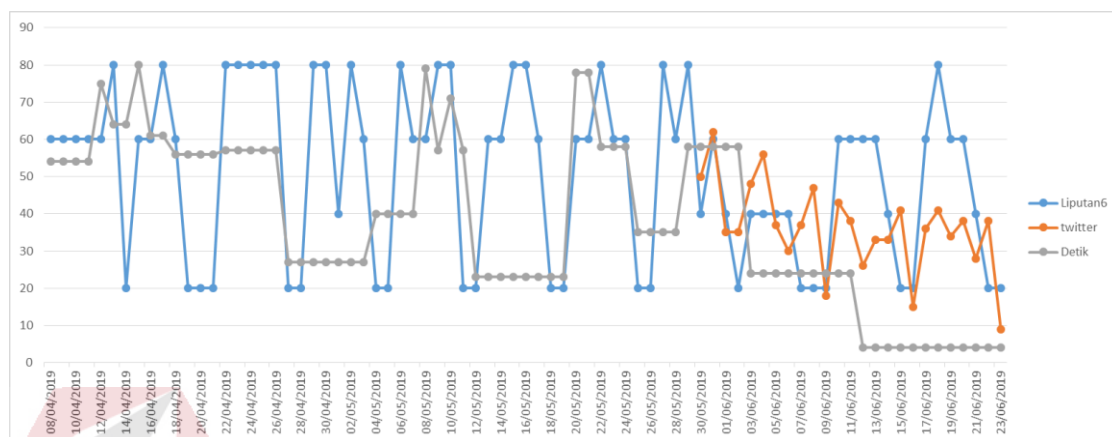


Gambar 4.8 Data *Scrapping* Twitter Sebelum 29 Mei

Untuk *scrapping* data dari media berita online dilakukan dengan mengambil semua berita yang ada pada tanggal yang ditentukan Pada setiap sumber. Dalam pengambilan data dari sumber Detik.com pada beberapa kesempatan akan mengalami kegagalan yang berasal dari sumber situs berita itu sendiri. Hal ini bukan disebabkan kesalahan dalam aplikasi, tetapi disebabkan oleh pencarian indeks berita berdasarkan tanggal yang beberapa kali tidak bisa digunakan. Hal ini dibuktikan dengan penggunaan fitur secara manual melalui internet *browser* .

Untuk detail banyak data yang didapat dari sumber perharinya dapat dilihat pada gambar 4.9. Seperti yang ditunjukkan pada gambar tersebut data yang diperoleh dari twitter sangatlah banyak sebelum 29 Mei. Hal ini menyebabkan aplikasi sering mencapai batas pengambilan data yang ditetapkan twitter sehingga harus menunggu kurang lebih 15 menit sebelum melanjutkan pengambilan data. Selain itu saat waktu yang digunakan untuk memproses data pun juga lebih lama,

karena itulah penulis memutuskan mengubah kata kunci pencarian *tweet*. Setelah dirubah banyak data yang didapat rata rata sama dengan data yang didapat dari sumber media berita online.



Gambar 4.9 Banyak Data Scrapping Tiap Hari

4.3.2 Karakteristik Data Komoditas

Pada sumber twitter ditemukan kendala pada proses pembersihan *emoticon*, yang belum dapat bersih secara sempurna. Hal ini dapat diatasi dengan menambah list “emotes” yang berisi kode dari emoticon dan digunakan untuk mendeteksi kode emoticon yang ada pada *tweet*. Kendala ini tetap tidak mempengaruhi keakuratan pencarian data harga komoditas. Adapun detail perbandingan data yang memiliki harga komoditas setelah diproses adalah sebagai berikut:

Tabel 4.8 Jumlah Data Harga Komoditas Yang Didapat

No	Nama Sumber	Data Mentah	Data Yang Membicarakan Komoditas	Data Yang Dapat Diambil Aplikasi	Akurasi(%)
1	Twitter	92204	97	51	52
2	Detik	2903	65	37	56
3	Liputan 6	3900	46	15	32
				Rata-Rata Akurasi(%)	47

Melihat data diatas dapat dilihat bahwa sumber yang paling banyak membicarakan data komoditas adalah sumber Twitter yang disusul dengan sumber Detik dan Liputan 6.

Tingkat akurasi rata-rata aplikasi ditemukan hanya mencapai 47 %, hal ini dikarenakan aplikasi hanya mencari nama komoditas pangan dan harga dari komoditas tersebut pada suatu teks. Sehingga faktor faktor lain seperti apakah komoditas itu barang import tidak dipertimbangkan. Harga yang diambil juga tidak melihat satuan berat yang tertera seperti kilogram atau ons. Selain itu memperluas knowledge base cek_tweet juga dapat meningkatkan tingkat akurasi aplikasi.

Adapun detail penyebaran data perkomoditas yang didapatkan dari proses seleksi pada setiap sumber data dapat dilihat pada tabel tabel dibawah ini:

Tabel 4.9 Detail Penyebaran Data Perkomoditas Twitter

No	Komoditas	Banyak Data	Data Yang Dapat Diambil Aplikasi
1	Daging sapi	26	16
2	Jeruk	24	8
3	Jagung	9	5
4	Tomat	7	4
5	Talas	6	3

No	Komoditas	Banyak Data	Data Yang Dapat Diambil Aplikasi
6	Bayam	5	3
7	Semangka	4	3
8	Kangkung	3	2
9	Rambutan	3	1
10	Tepung terigu	3	1
11	Daging ayam ras	2	2
12	Ketela pohon	2	0
13	Daging ayam kampung	1	1
14	Telur ayam ras	1	1
15	Telur puyuh	1	1
16	Ketela rambat	0	0
17	Telur ayam kampung	0	0

Tabel 4.10 Detail Penyebaran Data Perkomoditas Detik

No	Komoditas	Banyak Data	Data Yang Dapat Diambil Aplikasi
1	Semangka	17	4
2	Bayam	13	3
3	Daging sapi	11	10
4	Telur ayam ras	10	8
5	Daging ayam ras	9	8
6	Jagung	3	2
7	Tomat	2	2
8	Daging ayam kampung	0	0
9	Jeruk	0	0
10	Kangkung	0	0
11	Ketela rambat	0	0
12	Rambutan	0	0
13	Ketela pohon	0	0
14	Talas	0	0
15	Telur ayam kampung	0	0
16	Telur puyuh	0	0
17	Tepung terigu	0	0

Tabel 4.11 Detail Penyebaran Data Perkomoditas Liputan 6

No	Komoditas	Banyak Data	Data Yang Dapat Diambil Aplikasi
1	Semangka	18	0
2	Bayam	12	4
3	Jagung	3	1
4	Rambutan	3	1
5	Telur puyuh	3	3
6	Tomat	3	2
7	Telur ayam ras	2	2
8	Daging sapi	1	1
9	Telur ayam kampung	1	1
10	Daging ayam ras	0	0
11	Daging ayam kampung	0	0
12	Jeruk	0	0
13	Kangkung	0	0
14	Ketela rambat	0	0
15	Ketela pohon	0	0
16	Talas	0	0
17	Tepung terigu	0	0

Dari data penyebaran diatas dapat disimpulkan bahwa setiap sumber yang ada membicarakan komoditas pangan yang berbeda, sehingga fluktuasi harga komoditas pangan tidak dapat dilihat dari satu sumber saja.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan evaluasi hasil dari implementasi penelitian ini, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Aplikasi dapat memvisualisasikan fluktuasi harga komoditas pangan yang diambil dari Twitter, Liputan6, dan Detik.com, serta menampilkan data dari sumber pembanding PIHPS. Visualisasi juga dapat menggunakan jangka waktu yang diinginkan pengguna.
2. Dari data yang telah dikoleksi selama 77 hari, ditemukan bahwa sumber dengan data mentah dan data yang membicarakan nama komoditas terbanyak adalah sumber Twitter, dengan data mentah sebanyak 92204, dan data terseleksi sejumlah 97.
3. Penerapan seleksi data harga komoditas menggunakan algoritma Jaro-Winkler dalam aplikasi masih belum optimal. Dari hasil uji akurasi terlihat bahwa tingkat akurasi aplikasi dalam menyeleksi data harga komoditas pangan adalah 47%. Sumber dengan tingkat akurasi tertinggi adalah sumber detik dengan tingkat akurasi 56%. Hal ini terjadi karena aplikasi hanya mencari nama komoditas pangan dan harga dari komoditas tersebut pada suatu teks. Sehingga faktor faktor lain seperti apakah komoditas itu barang import tidak dipertimbangkan. Harga yang diambil juga tidak melihat satuan berat yang tertera seperti kilogram atau ons.

4. Dari data yang didapat pada sumber data Twitter, Detik.com, dan Liputan6, dapat disimpulkan bahwa tiap sumber membicarakan konten yang berbeda sehingga fluktuasi harga tidak dapat dilihat dari satu sumber saja.
5. Menurut pengujian dengan kurun waktu yang ditentukan, sumber Detik.com merupakan sumber yang paling banyak ditemukan data harga pada kurun waktu tersebut.

5.2 Saran

Adapun saran-saran yang dapat menjadi masukan bagi penelitian-penelitian dimasa depan antara lain adalah:

1. Mengembangkan algoritma untuk menyaring data harga yang memiliki kata 'ribu' didalamnya, serta dapat menyeleksi data satuan berat seperti "kg" agar harga komoditas yang didapat sama satuan beratnya. Selain itu juga mengembangkan algoritma yang dapat mengetahui maksud dari kalimat sehingga data yang didapat lebih akurat.
2. Meningkatkan *knowledge base* seperti database 'cek_tweet' dan *emotes* ketika mengambil data dari media sosial dan untuk meningkatkan pembersihan kode *emoticon*

DAFTAR PUSTAKA

- Asean Up. (2017, September 16). *Internet and digital landscape in Indonesia*. Retrieved from Asean Up: <https://aseanup.com/internet-digital-landscape-indonesia/>
- Bohang, F. K. (2018, Februari 22). *Berapa Jumlah Pengguna Internet Indonesia?* Retrieved from Kompas.com: <https://tekno.kompas.com/read/2018/02/22/16453177/berapa-jumlah-pengguna-internet-indonesia>
- BPS. (2019). *Badan Pusat Statistik*. Retrieved from Badan Pusat Statistik: <https://www.bps.go.id/menu/1/sejarah.html#masterMenuTab1>
- Dataflair. (2018, Februari 27). *What is Text Mining in Data Mining – Process & Applications*. Retrieved from Data-flair: <https://data-flair.training/blogs/text-mining/>
- Djuandi, F. (2002). *SQL Server 2000 untuk Profesional*. Jakarta: Elex Media Komputindo .
- Galetto, M. (2015, Desember 11). *What Is Data Mining?* Retrieved from ngdata: <https://www.ngdata.com/what-is-data-mining/>
- Ismail, E. (2018, November 8). *Kementan Adaptasi dan Mitigasi Hadapi Perubahan Iklim*. Retrieved from republika.co.id: <https://republika.co.id/berita/ekonomi/pertanian/18/11/08/phvujj453-kementan-adaptasi-dan-mitigasi-hadapi-perubahan-iklim>
- Kressa, A. V. (2019). *Rancang Bangun Portal Beasiswa Berbasis Web Menggunakan Metode Jaro-Winkler*. Surabaya: Institut Bisnis dan Informatika Stikom Surabaya.
- Kurniawati, A., Puspitodjati, S., & Rahman, S. (2010). *mplementasi Algoritma Jaro-Winkler Distance untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia*. Depok: Universitas Gunadarma.
- Kusrini, A. K. (2007). *Tuntunan Praktis Membangun Sistem Informasi Akuntansi Dengan Visual Basic Dan Microsoft SQL Server*. Yogyakarta: Andi Offset.
- Liputan6. (2019). *Tentang Kami Liputan6.com*. Retrieved from Liputan6: <https://www.liputan6.com/info/tentang-kami>
- Lutz, M. (2014). *Python Pocket Reference, Fifth Edition*. Sebastopol: O'Reilly.
- Mindfire Solutions. (2017, April 23). *Advantages and Disadvantages of Python Programming Language*. Retrieved from Medium:

<https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121>

- Moufarrege, S. (2015, Maret 31). *Advantage & Disadvantages of Microsoft SQL / Techwalla.com*. Retrieved from Techwalla: <https://www.techwalla.com/articles/advantages-disadvantages-of-microsoft-sql>
- Pressman, R. (2015). *Pendekatan Praktisi Buku 1*. Yogyakarta: Penerbit Andi.
- Ratte, J. B. (2016, Maret 23). *pyjarowinkler .PyPl*. Retrieved from Python Software Foundation: <https://pypi.org/project/pyjarowinkler/>
- Richardson, L. (2015). *Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation*. Retrieved from Crummy: The Site: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- Ridgen, J. (2018, 01 23). *Tweepy: a Python Library for the Twitter API*. Retrieved from Medium: <https://medium.com/@jasonrigden/tweepy-a-python-library-for-the-twitter-api-9d0537dcebd4>
- Rouse, M. (2018, April 4). *What is text mining(text analysis)?* Retrieved from Search Business Analytics: <https://searchbusinessanalytics.techtarget.com/definition/text-mining>
- Saifuloh, E. A. (2017). *Implementasi Natural Language Processing Untuk Mengurangi Risiko Terabaikannya Calon Pelanggan Pada PT. Shafira Tour & Travel*. Surabaya: Institut Bisnis dan Informatika Stikom Surabaya.
- Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-computer-interaction, Volume 85*. Boston: Addison Wesley Longman.
- Sommerville, I. (2011). *Software Engineering(Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.
- Wardani, F. K. (2019). *Analisis Sentimen Untuk Pemeringkatan Popularitas Situs Belanja Online Di Indonesia Menggunakan Metode Naive Bayes(Studi Kasus Data Sekunder)*. Surabaya: Institut Bisnis dan Informatika Stikom Surabaya.
- Waryanto, B., & Susanti, A. A. (2017). *Statistik Pertanian Agricultural Statistic 2017*. Jakarta: Pusat Data dan Informasi Pertanian Kementerian Pertanian Republik Indonesia.