

Chrome Extension untuk Data Grabber Media Sosial Twitter dengan Metode XPath Selector

Chrome Extension for Twitter Data Grabber using XPath Selector Method

Erinda Resha A¹, Antonius Rachmat C², Yuan Lukito³

^{1,2,3}Informatika, Universitas Kristen Duta Wacana

E-mail : ¹erinda.resha@ti.ukdw.ac.id, ²anton@ti.ukdw.ac.id, ³yuanlukito@ti.ukdw.ac.id

Abstrak

Kebutuhan *dataset* sebagai sumber penelitian untuk analisis oleh data *scientist* terus meningkat seiring dengan semakin meningkatnya rasa keingintahuan manusia. Misalnya seperti suatu perusahaan ingin mengetahui tanggapan konsumennya tentang produk yang baru saja diluncurkan, sehingga mereka dapat menyusun langkah selanjutnya yang akan diambil oleh perusahaan tersebut. Tentu saja untuk mengetahui hal tersebut, perlu dilakukan analisis terhadap tanggapan konsumen terlebih dahulu. Analisis dilakukan menggunakan sumber data yang dapat ditemukan salah satunya pada sosial media seperti Twitter. Penelitian ini dimaksudkan untuk menyediakan aplikasi yang dapat digunakan menjadi data *grabber* untuk mengumpulkan data dari media sosial Twitter. Aplikasi data *grabber* dirancang sebagai *Chrome Extension* dengan implementasi metode *XPath Selector* diharapkan dapat mengumpulkan data tanpa terbatas oleh batasan akses API Twitter. Sebagai gantinya dengan menerapkan metode ini, aplikasi perlu mengakses halaman web Twitter. Pemasangan aplikasi sebagai *Chrome Extension* memudahkan untuk menjalankan aplikasi sembari mengakses halaman web Twitter pada browser. Aplikasi TweetGrabberERA berhasil dibangun pada penelitian ini sesuai kebutuhan yang diharapkan. Pengujian yang dirancang telah menunjukkan bahwa aplikasi data *grabber* dengan mengimplementasikan metode *XPath Selector* berjalan dengan baik dan membuktikan bahwa aplikasi dapat menghasilkan *dataset* yang untuk keperluan analisis oleh data *scientist*.

Kata kunci: *XPath Selector*, *Chrome Extension*, data *grabber*, Twitter

Abstract

The need of datasets as research sources for analysis by data scientists continues to increase along with the increasing of human curiosity. For example, a company wants to know consumers' response about the products that have been launched, so they can arrange the next steps to be taken by the company. To do so, analysis of consumers' response must be done. The analysis can be done using data sources that come from social media such as Twitter. This research is intended to provide an application used as a data grabber to collect data from social media Twitter. This application is designed as a Chrome Extension with implementation of XPath Selector expected to collect data without being limited by Twitter API access restrictions. Instead of applying this method, this application needs to access Twitter web page. As Chrome Extension, it's easy to run while accessing Twitter web pages in the browser. TweetGrabberERA was successfully built in this study. It has shown that data grabber application with implementation of the XPath Selector runs well and proved can produce a dataset for the purposes of analysis by data scientists.

Keywords: *XPath Selector*, *Chrome Extension*, data *grabber*, Twitter

1. PENDAHULUAN

Media sosial digunakan sebagai media komunikasi dan berbagi informasi secara daring. Media sosial telah menjadi salah satu kebutuhan hidup bagi kaum muda maupun tua sehingga kepopuleran media sosial terus meningkat dari tahun ke tahun. Salah satu media sosial yang masih menjadi tren yaitu Twitter. Twitter merupakan media sosial daring yang cukup populer, hingga saat ini tercatat 330 juta pengguna aktif tiap bulannya [1]. Twitter menampilkan *tweet* yaitu

kiriman yang memiliki batas tidak lebih dari 280 karakter [2], membuatnya terlihat sederhana sehingga mudah untuk dibaca. Informasi-informasi yang banyak ditemukan biasanya berupa berita, ilmu pengetahuan, opini, dan masalah personal. Agar dapat memanfaatkan informasi yang ada, perlunya sebuah aplikasi yang dapat membantu mengumpulkan informasi-informasi tersebut dalam suatu bentuk *dataset* sehingga pengolahan informasi akan menjadi lebih mudah.

Data *grabber* merupakan salah satu solusi, dimana aplikasi tersebut dapat bekerja untuk mengambil data secara otomatis. Metode yang mudah dijumpai untuk pengambilan data pada Twitter adalah melalui API Twitter seperti yang telah diterapkan pada Tweepy. Tweepy sendiri merupakan sebuah *library* di Python yang digunakan untuk mengakses API Twitter. Sayangnya untuk mengakses API Twitter, perlu melakukan pengesahan kredensial dimana pengguna akan diminta untuk memasukkan *consumer key*, *consumer secret key*, *access token key*, *access token secret key* yang hanya didapat melalui izin dari Twitter. Hal tersebut sungguh membatasi pengguna untuk dapat mengumpulkan *tweet*.

Untuk memudahkan penggunaan aplikasi, browser Chrome saat ini telah dilengkapi dengan *add-on* yang disebut Chrome *Extension*. Dengan adanya Chrome *Extension* pengguna dapat melakukan kustomisasi terhadap browsernya sehingga akan mendapatkan pengalaman baru yang berbeda [3]. Memasang Chrome *Extension* pada browser Chrome, penggunaan aplikasi sambil mengakses situs web pada browser menjadi lebih sederhana.

Rizaldi dan Arief [4] meneliti tentang perbandingan metode web *scraping* menggunakan CSS *Selector* dan XPath *Selector*. Penelitian ini membahas permasalahan bagaimana kelengkapan data yang diperoleh, penggunaan memori, dan kecepatan waktu yang diperlukan ketika menggunakan XPath *Selector* dan CSS *Selector* pada situs berita sebagai metode web *scraping* menggunakan tools Scrapy. Kesimpulan dari penelitian mereka adalah penggunaan metode XPath *Selector* menghasilkan artikel yang lebih lengkap dibandingkan dengan menggunakan metode CSS *Selector* untuk web *scraping* situs berita walaupun membutuhkan proses lain untuk menghilangkan kode HTML yang tidak diinginkan. Penggunaan memori oleh metode XPath *Selector* dan CSS *Selector* tidak memiliki perbedaan yang signifikan bahkan cenderung sama sedangkan pada sisi kecepatan waktu secara umum metode XPath *Selector* memiliki waktu proses yang lebih cepat daripada menggunakan metode CSS *Selector*.

Adapun masalah yang akan dihadapi ketika menggunakan metode XPath *Selector*. Struktur web seperti media sosial Twitter ini akan selalu diperbarui dalam jangka waktu tertentu. Hal tersebut jelas akan menimbulkan kendala ketika menggunakan XPath *Selector* yang bekerja dengan cara melakukan navigasi pada halaman HTML [5]. Perlunya penyesuaian kembali agar aplikasi tetap berjalan sesuai harapan.

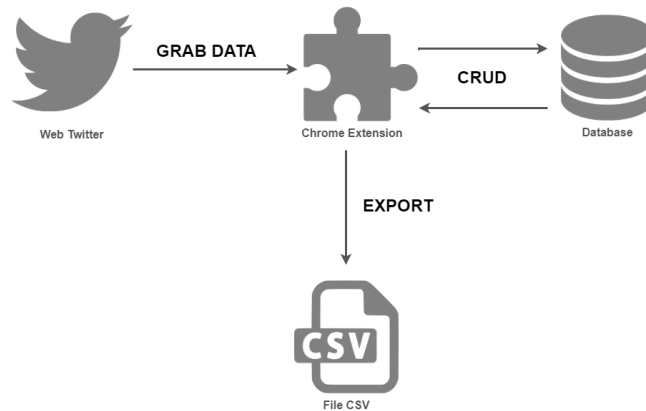
Penelitian ini bertumpu pada penelitian yang dilakukan oleh Rizaldi dan Arief. Penelitian ini akan membuat aplikasi data *grabber* dalam bentuk *add-on* pada browser Chrome yang disebut Chrome *Extension*. Aplikasi dibangun untuk menjadi data *grabber* media sosial Twitter dengan mengimplementasikan metode XPath *Selector*.

2. METODE PENELITIAN

2.1. Perancangan Sistem

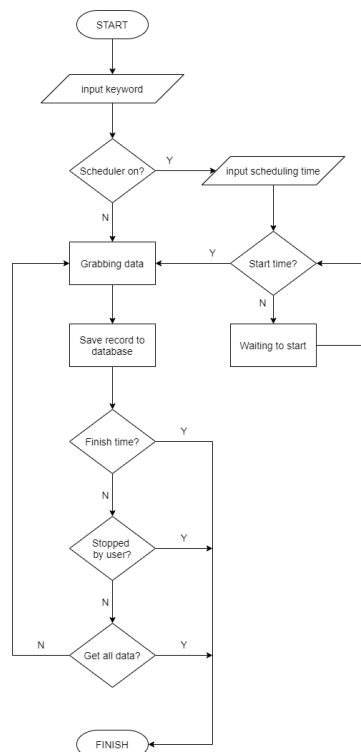
Aplikasi Chrome *Extension* untuk Data *Grabber* Media Sosial Twitter dengan metode XPath *Selector* disusun dengan arsitektur seperti pada Gambar 1. Aplikasi yang berupa Chrome *Extension* mengambil data dari situs media sosial Twitter yang dibangun menggunakan JavaScript. Pengambilan data dilakukan melalui elemen HTML situs media sosial Twitter dengan mencari XPath yang tepat, sehingga tidak memerlukan API dari Twitter. Data yang telah diperoleh selanjutnya akan diolah, disusun, lalu disimpan ke dalam penyimpanan lokal. IndexedDB merupakan *database* lokal pada browser yang digunakan oleh aplikasi. Dengan adanya AlaSQL, IndexedDB akan lebih mudah diakses karena AlaSQL telah menyediakan *library* yang memberi kenyamanan akses penuh melalui JavaScript ke *database* termasuk

IndexedDB menggunakan *query* SQL. Data yang telah terkumpul dapat diekspor oleh aplikasi menjadi *file* dalam bentuk CSV.



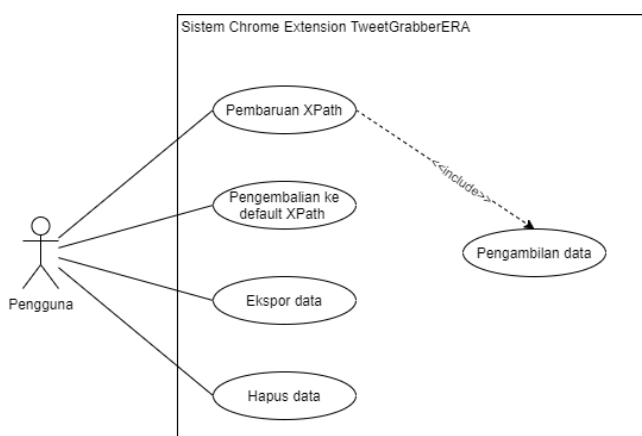
Gambar 1. Blok diagram sistem aplikasi

Pada alur kerja sistem untuk *Chrome Extension* ini, pengguna diwajibkan untuk mengisi input yang diminta sebelum aplikasi dijalankan. Fitur penjadwalan merupakan fitur opsional untuk digunakan. *Flowchart* untuk alur aplikasi dijelaskan pada Gambar 2.



Gambar 2. *Flowchart* Chrome Extension

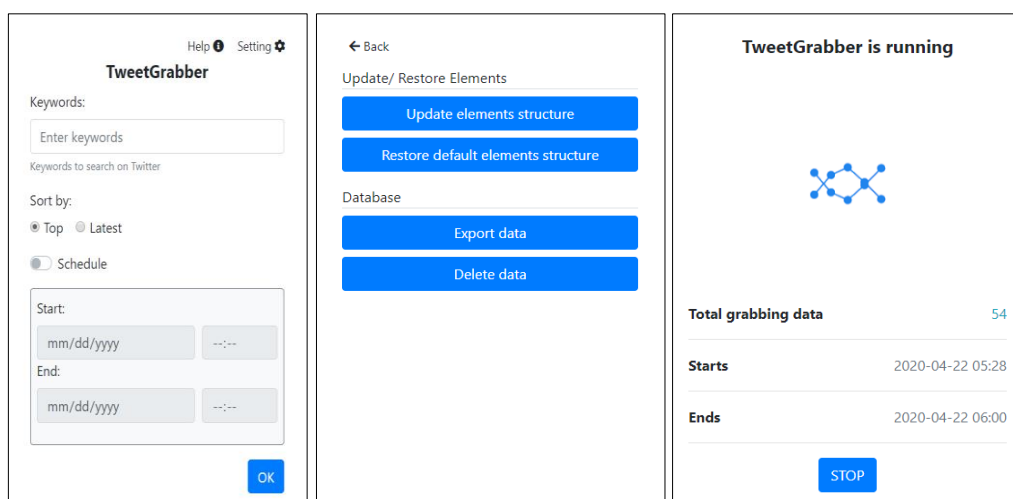
Untuk memulai aplikasi, pengguna perlu untuk melakukan beberapa persiapan seperti menginputkan *keyword* dan memilih ingin menggunakan fitur penjadwalan atau tidak. Jika fitur penjadwalan dinyalakan, maka pengguna harus menginputkan waktu mulai dan selesai. Aplikasi mulai bekerja ketika pengguna telah menekan tombol OK. Aplikasi melakukan pengecekan apakah pengguna menggunakan fitur penjadwalan atau tidak. Jika fitur penjadwalan dijalankan, maka aplikasi akan mulai dan mengakhiri kerja *grabbing* data sesuai jadwal yang diinputkan. Jika fitur penjadwalan tidak dijalankan, maka aplikasi langsung melakukan *grabbing* data saat itu juga dan akan berakhir ketika pengguna menekan tombol STOP.



Gambar 3. Use case diagram sistem

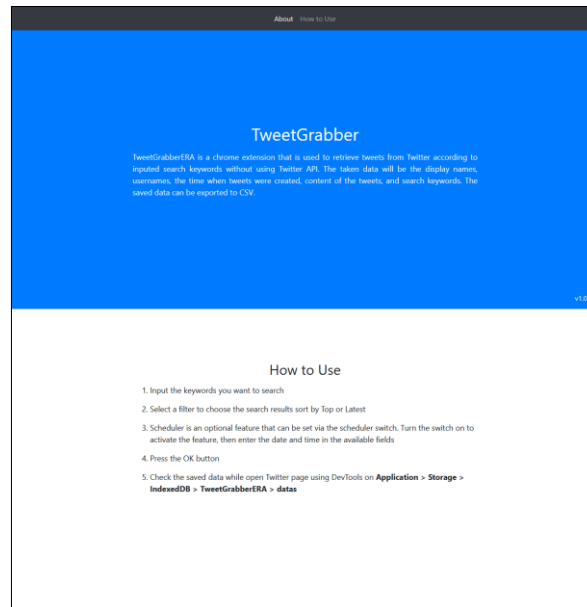
Sesuai dengan *use case* diagram pada Gambar 3 pengguna dapat melakukan beberapa hal seperti memperbarui XPath, mengembalikan XPath ke *default* XPath, melakukan ekspor data, dan menghapus data. Untuk melakukan pembaruan XPath, pengguna setidaknya harus telah melakukan fitur pengambilan data. Hal ini diperlukan agar aplikasi dapat mendeteksi adanya perubahan pada struktur elemen halaman web Twitter atau tidak.

Rancangan halaman antarmuka aplikasi Chrome *Extension* terdiri dari 2 bagian, yaitu halaman *pop up* dan halaman tambahan (*additional page*). Halaman *pop up* akan muncul ketika ikon Chrome *Extension* diklik untuk mengaktifkan aplikasi. Saat Chrome *Extension* belum berjalan, *pop up* akan meminta pengguna untuk melakukan setting awal seperti memasukkan *keyword*, memilih urutan tampilan *tweet*, dan membuat penjadwalan kerja aplikasi. Selain itu ada halaman *setting* untuk melakukan pengaturan aplikasi seperti *scan* XPath elemen web Twitter, mengembalikan tatanan XPath aplikasi ke bentuk awal yaitu Twitter versi rilis 21 April 2020, dan melakukan *export*, atau *delete database*. Tampilan halaman *pop up*, dapat dilihat pada Gambar 4.



Gambar 4. Halaman *pop up* menu utama, menu *setting*, dan *pop up* ketika aplikasi berjalan

Sebuah halaman yang berisi deskripsi aplikasi, penjelasan cara kerja aplikasi secara garis besar, dan kolom bantuan. Agar nantinya dapat diakses oleh pengguna jika mengalami kesulitan ketika menjalankan aplikasi. Rupa desain tampilan dapat dilihat pada Gambar 5.



Gambar 5. Halaman tambahan berisi informasi aplikasi

Basis data yang digunakan pada aplikasi Chrome *Extension* dibuat menggunakan IndexedDB. Aplikasi hanya memiliki satu *objectStore* bernama *datas* dengan struktur seperti pada Tabel 1.

Tabel 1 Struktur Tabel Database TweetGrabberERA

Tabel Datas		
Key	Value	Keterangan
tweet_id	String	id <i>tweet</i>
created_at	Date	waktu dan tanggal <i>tweet</i> dibuat
name	String	nama akun pembuat <i>tweet</i>
username	String	username pembuat <i>tweet</i>
text_content	String	isi teks <i>tweet</i> kiriman pengguna
keyword	String	kata yang dicari dalam <i>tweet</i>

2.2 Perancangan Pengujian Sistem

2.2.1 Pengujian Aplikasi

Pengujian aplikasi akan berfokus tentang kesesuaian kerja aplikasi dengan rancangan yang telah dibuat. Penulis telah membuat beberapa skenario-skenario kerja seluruh fitur aplikasi untuk melakukan pengujian ini. Pengujian terhadap aplikasi ini dijelaskan melalui Tabel 2.

Aplikasi TweetGrabberERA yang telah diuji selanjutnya akan dibandingkan dengan aplikasi pada penelitian terdahulu, yaitu *Automatic Crawling Data in Twitter* yang dibangun oleh Sembodo dkk. [6]. Hal ini dilakukan untuk menunjukkan kelebihan dan kekurangan berdasarkan fitur dari aplikasi atau program yang memiliki fungsi yang sama yaitu dapat melakukan pengambilan data pada Twitter.

Tabel 2 Skenario Pengujian

Fitur <i>Grabbing Data</i>		
ID	Skenario Test	Expected Results
TGF01001	Aplikasi meminta pengguna untuk menginputkan <i>keyword</i>	Muncul peringatan yang meminta pengguna untuk menginputkan <i>keyword</i> .
TGF01002	Aplikasi mengambil data dari halaman web Twitter yang melakukan pencarian mengandung <i>keyword</i> "morning" dengan filter "Top"	Data yang mengandung <i>keyword</i> "morning" dengan urutan sesuai filter "Top" tersimpan dalam <i>database</i> dan halaman <i>pop up</i> menampilkan total seluruh data yang telah tersimpan dalam <i>database</i> .

TGF01003	Aplikasi mengambil data dari halaman web Twitter yang melakukan pencarian mengandung keyword "morning" dengan filter "Latest"	Data yang mengandung keyword "morning" dengan urutan sesuai filter "Latest" tersimpan dalam database dan halaman pop up menampilkan total seluruh data yang telah tersimpan dalam database.
TGF01004	Aplikasi melakukan scroll pada halaman web Twitter ketika belum mencapai hasil pencarian paling akhir	Halaman web Twitter melakukan scroll secara otomatis.
TGF01005	Aplikasi berhenti mengambil data ketika sampai pada hasil pencarian paling akhir	Aplikasi berhenti bekerja lalu mengeluarkan pop up bahwa aplikasi telah berhenti karena telah sampai pada hasil pencarian paling akhir.
TGF01006	Aplikasi berhenti bekerja ketika hasil pencarian sesuai keyword "zxcvbnmtgb" dan filter "Top" dari pengguna tidak ditemukan	Aplikasi berhenti bekerja lalu mengeluarkan pop up bahwa aplikasi telah berhenti karena hasil pencarian tidak ditemukan.
TGF01007	Aplikasi berhenti bekerja ketika hasil pencarian tidak muncul setelah 1 menit berlalu	Aplikasi berhenti bekerja lalu mengeluarkan pop up yang meminta pengguna untuk mengecek koneksi internet.
TGF01008	Aplikasi berhenti bekerja ketika URL halaman pada tab browser yang digunakan untuk mengambil data diubah	Aplikasi berhenti bekerja lalu mengeluarkan pop up bahwa aplikasi telah berhenti karena URL halaman telah diubah.
TGF01009	Aplikasi berhenti bekerja ketika tab browser yang digunakan untuk mengambil data ditutup	Aplikasi berhenti bekerja
TGF01010	Aplikasi berhenti bekerja ketika jendela browser yang digunakan untuk mengambil data ditutup	Aplikasi berhenti bekerja.
TGF01011	Pengetesan ulang TGF01001 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01001
TGF01012	Pengetesan ulang TGF01002 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01002
TGF01013	Pengetesan ulang TGF01003 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01003
TGF01014	Pengetesan ulang TGF01004 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01004
TGF01015	Pengetesan ulang TGF01005 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01005
TGF01016	Pengetesan ulang TGF01006 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01006
TGF01017	Pengetesan ulang TGF01007 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01007
TGF01018	Pengetesan ulang TGF01008 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01008
TGF01019	Pengetesan ulang TGF01009 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01009
TGF01020	Pengetesan ulang TGF01010 setelah melakukan update XPath	Sama seperti hasil pengetesan TGF01010
Fitur Penjadwalan		
ID	Skenario Test	Expected Results
TGF02001	Aplikasi berjalan sesuai jadwal yang diinputkan pada kolom Start	Muncul tanggal dan waktu sesuai input dari pengguna pada kolom Start di pop up. Aplikasi mulai bekerja dengan menunjukkan gambar loading bergerak dan jumlah data bertambah ketika tanggal dan waktu saat ini sudah sesuai seperti pada kolom Start.
TGF02002	Aplikasi berjalan sesuai jadwal yang diinputkan pada kolom End	Muncul tanggal dan waktu sesuai input dari pengguna pada kolom End di pop up. Aplikasi langsung bekerja dan akan berhenti ketika tanggal dan waktu saat ini sudah sesuai seperti pada kolom End
TGF02003	Aplikasi berjalan sesuai jadwal yang diinputkan pada kolom Start dan End	Muncul tanggal dan waktu sesuai input dari pengguna pada kolom Start dan End di pop up. Aplikasi mulai bekerja dengan menunjukkan gambar loading bergerak dan jumlah data bertambah ketika tanggal dan waktu saat ini sudah sesuai seperti pada kolom Start. Aplikasi akan berhenti ketika tanggal dan waktu saat ini sudah sesuai seperti pada kolom End
TGF02004	Aplikasi meminta pengguna untuk menginputkan penjadwalan	Muncul pop up bahwa pengguna harus menginputkan waktu tanggal dan waktu pada kolom Start atau End
TGF02005	Aplikasi memperingatkan pengguna bahwa jadwal pada kolom Start tidak valid karena tanggal tidak tepat	Muncul pop up bahwa tanggal atau waktu pada kolom Start tidak valid
TGF02006	Aplikasi memperingatkan pengguna bahwa jadwal pada kolom Start tidak valid karena waktu tidak tepat	Muncul pop up bahwa tanggal atau waktu pada kolom Start tidak valid
TGF02007	Aplikasi memperingatkan pengguna bahwa jadwal pada kolom End tidak valid karena tanggal tidak tepat	Muncul pop up bahwa tanggal atau waktu pada kolom End tidak valid

TGF02008	Aplikasi memperingatkan pengguna bahwa jadwal pada kolom <i>End</i> tidak valid karena waktu tidak tepat	Muncul <i>pop up</i> bahwa tanggal atau waktu pada kolom <i>End</i> tidak valid
TGF02009	Aplikasi memperingatkan pengguna bahwa jadwal pada kolom <i>Start</i> tidak valid karena tanggal dan waktu yang diinputkan merupakan tanggal dan waktu sebelum dari saat ini	Muncul <i>pop up</i> bahwa tanggal atau waktu pada kolom <i>Start</i> tidak boleh sebelum dari saat ini
TGF02010	Aplikasi memperingatkan pengguna bahwa jadwal pada kolom <i>End</i> tidak valid karena tanggal dan waktu yang diinputkan merupakan tanggal dan waktu sebelum dari saat ini	Muncul <i>pop up</i> bahwa tanggal atau waktu pada kolom <i>End</i> tidak boleh sebelum dari saat ini
TGF02011	Aplikasi memperingatkan pengguna bahwa jadwal pada kolom <i>Start</i> harus sebelum jadwal pada kolom <i>End</i>	Muncul <i>pop up</i> bahwa tanggal atau waktu pada kolom <i>Start</i> harus sebelum dari tanggal atau waktu pada kolom <i>End</i>
Fitur Pengelolaan Xpath		
ID	Skenario Test	Expected Results
TGF03001	Aplikasi melakukan <i>update</i> XPath pada halaman Twitter versi rilis 21 April 2020	Tombol <i>update</i> tidak dapat ditekan dan <i>update</i> XPath tidak dapat dilakukan
TGF03002	Aplikasi melakukan <i>restore</i> XPath sesuai <i>default</i>	XPath kembali ke versi rilis 21 April 2020
TGF03003	Aplikasi melakukan <i>update</i> XPath pada halaman Twitter selain versi rilis 21 April 2020	XPath telah mengalami <i>update</i>
TGF03004	Aplikasi melakukan <i>update</i> XPath pada halaman Twitter selain versi rilis 21 April 2020 dan diinterupsi oleh pengguna	Muncul <i>pop up</i> bahwa XPath gagal dilakukan <i>update</i> dan XPath kembali ke versi rilis 21 April 2020
TGF03005	Aplikasi melakukan <i>update</i> XPath kembali pada halaman Twitter selain versi rilis 21 April 2020 setelah <i>update</i> XPath sebelumnya diinterupsi oleh pengguna	XPath telah mengalami <i>update</i>
Fitur Pengelolaan Database		
ID	Skenario Test	Expected Results
TGF04001	Aplikasi mengekspor data yang tersedia pada <i>database</i>	<i>File</i> TweetGrabberERA.csv terunduh sesuai data pada <i>database</i>
TGF04002	Aplikasi menghapus data pada <i>database</i>	<i>Database</i> TweetGrabberERA pada IndexedDB terhapus
TGF04003	Aplikasi mengekspor data setelah <i>database</i> dihapus	<i>File</i> TweetGrabberERA.csv terunduh tanpa terisi data apapun.

2.2.2 Pengujian Validasi Dataset

Pengujian validasi *dataset* dilakukan oleh penulis akan dilalui dengan langkah-langkah yang telah disusun ke dalam beberapa skenario. Penulis akan melakukan validasi terhadap *file* CSV yang disiapkan dari hasil ekspor aplikasi yang berisi 15.000 rekam data. Validasi terhadap *file* CSV dilakukan dengan menggunakan *tool open source* CSV Validator GUI 1.2 oleh *The National Archives*. *File* CSV Schema pada Gambar 6 yang disusun dalam bahasa skema berbasis teks diperlukan untuk validasi sebagai *file* yang mendeskripsikan isi data pada *file* CSV. Ekspektasi dari pengujian ini, diharapkan *file* CSV mendapatkan respons *pass* dari validator.

```

version 1.0
@totalColumns 6
tweet_id: regex("[0-9]+")
created_at: regex("\d{4,4}-\d{2,2}-\d{2,2} \d{2,2}:\d{2,2}:\d{2,2}")
name:
username: notEmpty
text_content: notEmpty
keyword: notEmpty
    
```

Gambar 6. CSV Schema

Pengujian selanjutnya akan dilakukan dengan cara melakukan pengolahan data dengan *dataset* ini menggunakan *tools* Orange 3.24.0 yang merupakan *open source machine learning* dan visualisasi data. Dari 15.000 rekam data yang ada pada *file* CSV akan diambil sebanyak 150 *sample* data secara acak untuk dilakukan analisis sentimen dengan metode VADER (*Valence Aware Dictionary and sEntiment Reasoner*) melalui fitur yang telah tersedia pada *tools* Orange. Sebelum dilakukan analisis sentimen, data akan melewati tahap *preprocess text*. Ada 3 hal yang akan dilakukan pada tahap tersebut yaitu sebagai berikut:

- Transformasi untuk mengubah data pada setiap kata menjadi huruf kecil, menghapus aksen, dan menghapus URL yang tersimpan pada teks.

- Tokenisasi untuk memecah teks menjadi komponen yang lebih kecil yaitu potongan-potongan kata.
- Penyaringan untuk menghapus kata sambung dan tanda baca termasuk simbol *hashtag* dan *at*.

Analisis sentimen yang dihasilkan oleh *tools* Orange nantinya akan dibandingkan dengan analisis sentimen yang dihasilkan oleh pakar, untuk melihat apakah hasil dari *tools* dengan pakar memiliki jawaban yang sama atau tidak. Analisis sentimen oleh pakar didapatkan melalui kuesioner dengan kriteria responden yang dicari adalah berstatus sebagai seorang mahasiswa. Sentimen terbanyak dari analisis setiap rekam data yang dinyatakan oleh pakar akan digunakan untuk dibandingkan dengan hasil analisis sentimen dari *tool* Orange.

Menurut Albert dkk. [7] *confusion matrix* menggunakan data dalam bentuk matriks yang menyajikan informasi prediksi dan aktual dari suatu sistem klasifikasi yang memiliki data jawaban benar. Dari informasi pada matriks tersebut dapat diperoleh akurasi yaitu nilai seberapa akurat data terklasifikasi dengan benar.

		Predicted			
		A ₁	... A _j ...	A _n	
Actual	A ₁	N ₁₁	N _{1j}	N _{1n}	
	⋮	⋮	⋮	⋮	
	A _i	N _{i1}	N _{ij}	N _{in}	
	⋮	⋮	⋮	⋮	
A _n	N _{n1}	N _{nj}	N _{nn}		

Gambar 7. Confusion matrix untuk multi-class

Pada Gambar 7 menurut Deng dkk. [8] menampilkan bentuk *confusion matrix* untuk klasifikasi *multi-class* (A₁, A₂, hingga A_n), dengan keterangan N_{ij} sebenarnya merupakan kelas A_i tetapi diprediksi sebagai kelas A_j sehingga akurasi dirumuskan menjadi Rumus 1.

$$\text{Akurasi} = \frac{\sum_{i=1}^n N_{ii}}{\sum_{i=1}^n \sum_{j=1}^n N_{ij}} \quad (1)$$

Perhitungan akan digunakan sesuai dengan penelitian oleh Deng dkk. (2016) tetapi dalam persentase sehingga ditulis seperti Rumus 2.

$$\text{Akurasi} = \frac{\sum_{i=1}^n N_{ii}}{\sum_{i=1}^n \sum_{j=1}^n N_{ij}} \times 100\% \quad (2)$$

Hasil dari pengujian ini diharapkan dapat menunjukkan bahwa *dataset* yang didapat dari ekspor data aplikasi bisa digunakan untuk keperluan analisis oleh data *scientist* salah satunya untuk analisis sentimen.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

3.1.1 Fitur Grabbing Data

Grabbing data merupakan fitur utama aplikasi ini. Aplikasi akan bekerja melakukan pengambilan data *tweet* kiriman pengguna pada halaman web media sosial Twitter. Pengambilan data *tweet* dilakukan dengan cara menelusuri elemen-elemen pada halaman pencarian Twitter yang sedang dibuka pada browser. Halaman yang dibuka merupakan halaman pencarian Twitter dalam bahasa Inggris yang mengandung *keyword* yang dimasukkan oleh pengguna.

Halaman web Twitter menggunakan AJAX untuk menampilkan seluruh *tweet* di halaman web sehingga aplikasi perlu menunggu beberapa waktu hingga *tweet* ditampilkan sepenuhnya. Ketika halaman web Twitter telah tampil sepenuhnya, aplikasi akan mulai mengambil data sesuai XPath yang telah disimpan pada *local storage* oleh aplikasi. Dapat dilihat daftar XPath yang digunakan

oleh aplikasi pada Table VI. Seluruh *key* pada Tabel 6 memiliki *value* berupa XPath yang tersimpan pada *local storage* sesuai halaman web Twitter versi rilis 21 April 2020.

Tabel 6 Daftar Xpath Yang Tersimpan Pada Local Storage

<i>Key</i>	<i>Value</i>
xpathSearchKeyword	//input[@enterkeyhint='search']
xpathNoResults	//div[@data-testid='primaryColumn']/div[1]/div[2]/div[1]/div[1]/div[1]/span[1]
xpathMain	//section[@role='region'][1]/div[1]/div[1]/div[1]
xpathMinorName	//article[1]/div[1]/div[@data-testid='tweet']/div[2]/div[1]/div[1]/div[1]/div[1]/a[1]/div[1]/div[1]
xpathMinorUsername	//article[1]/div[1]/div[@data-testid='tweet']/div[2]/div[1]/div[1]/div[1]/div[1]/a[1]/div[1]/div[2]
xpathMinorFirstLine	//article[1]/div[1]/div[@data-testid='tweet']/div[2]/div[2]/div[1]
xpathMinorSecondLine	//article[1]/div[1]/div[@data-testid='tweet']/div[2]/div[2]/div[2]
xpathMinorTweetIdRaw	//article[1]/div[1]/div[@data-testid='tweet']/div[2]/div[1]/div[1]/div[1]/div[1]/a[1]
xpathMinorDate	//article[1]/div[1]/div[@data-testid='tweet']/div[2]/div[1]/div[1]/div[1]/div[1]/a[1]/time[1]

Masing-masing *key* pada Tabel 6 dimasukkan ke dalam variabel pada *code* aplikasi. Inisialisasi nama variabel pada *code* dibuat sama dengan nama *key* pada *local storage*. Variabel-variabel tersebut perlu disusun kembali pada *code* untuk mendapatkan data yang dibutuhkan untuk disimpan pada *database*. Penyesuaian susunan variabel-variabel pada *code* tersebut dijelaskan melalui Tabel 7.

Tabel 7 Daftar Variabel Yang Digunakan Untuk Menampung Xpath

<i>Variabel</i>	<i>Value</i>
contentKeyword	xpathSearchKeyword
contentName	xpathMain + "/div[" + tweetLine + "]" + xpathMinorName
contentUsername	xpathMain + "/div[" + tweetLine + "]" + xpathMinorUsername
contentFirstLine	xpathMain + "/div[" + tweetLine + "]" + xpathMinorFirstLine
contentSecondLine	xpathMain + "/div[" + tweetLine + "]" + xpathMinorSecondLine
contentTweetIdRaw	xpathMain + "/div[" + tweetLine + "]" + xpathMinorTweetIdRaw
contentDate	xpathMain + "/div[" + tweetLine + "]" + xpathMinorDate

Note. tweetLine = baris urutan tweet pada halaman web Twitter yang akan diambil

XPath yang telah disusun akan digunakan untuk mendapatkan elemen dari halaman web Twitter. Salah satu method dari DOM (*Document Object Model*) yaitu *document.evaluate()* berfungsi untuk mengevaluasi ekspresi XPath. Hasil pengembalian dari *method* tersebut dapat memiliki tipe yang bervariasi. Pada kasus ini penulis menggunakan properti *singleNodeValue* untuk mendapatkan hasil pengembalian dengan tipe elemen *node*. *Method* *document.evaluate()* telah penulis terapkan pada fungsi *getElementByXpath()* yang dapat dilihat pada Gambar 8.

```
function getElementByXpath(path) {
  return document.evaluate(path, document.body, null, XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
}
```

Gambar 8. Penerapan *method* *document.evaluate()*

Hasil pengembalian bisa *null* jika XPath yang dievaluasi tidak ada dalam dokumen HTML. Selanjutnya elemen yang berhasil didapat akan disaring kembali untuk diambil sesuai kebutuhan.

3.1.2 Fitur Penjadwalan

Penjadwalan menjadi fitur pilihan pada aplikasi ini. Pengguna dapat menggunakan fitur penjadwalan untuk mengatur kapan aplikasi mulai berjalan maupun berhenti. Fasilitas pada fitur

ini mengizinkan pengguna untuk dapat mengatur jadwal dengan memasukkan salah satu waktu aplikasi dimulai, waktu aplikasi selesai, atau keduanya sekaligus. Fitur penjadwalan dapat diaktifkan melalui *switch* pada halaman utama menu *pop up* oleh pengguna.

Aplikasi akan langsung melakukan *grabbing* data jika fitur penjadwalan tidak diaktifkan, dan akan berhenti ketika pengguna menekan tombol STOP pada Gambar 4. Jika fitur penjadwalan diaktifkan, maka program akan mulai melakukan *grabbing* data sesuai jadwal yang telah dimasukkan oleh pengguna.

Pengecekan waktu dilakukan di bagian *background* aplikasi. *Background* akan bekerja mengecek *datetime* setiap detik untuk melihat apakah sudah waktunya bagi aplikasi untuk mulai melakukan *grabbing* data atau melihat apakah sudah waktunya bagi aplikasi untuk menghentikan aplikasi sesuai *datetime* penjadwalan yang telah diatur. Tugas pengecekan waktu pada *background* ini hanya bekerja jika fitur penjadwalan diaktifkan. Jika tidak diaktifkan, maka *background* tidak perlu melakukan pengecekan waktu.

Tidak lepas dari kerja browser, fitur penjadwalan hanya dapat berjalan jika tab browser yang digunakan untuk mengambil data tetap dibiarkan terbuka. Jika tab browser atau jendela browser ditutup, maka aplikasi termasuk fitur penjadwalan akan berhenti bekerja.

3.1.3 Fitur Pengelolaan XPath untuk Update XPath

Fitur *update* XPath ini berfungsi untuk melakukan *update* XPath pada aplikasi jika terjadi perubahan pada struktur elemen halaman web Twitter. Tombol ini dapat ditemui pada menu *Setting* halaman *pop up* aplikasi. Secara *default* tombol tidak dapat diklik, tombol hanya bisa diklik apabila aplikasi mendeteksi adanya perubahan pada struktur elemen halaman web Twitter. Untuk dapat mendeteksi hal ini, pengecekan struktur elemen dilakukan oleh aplikasi setiap aplikasi memulai untuk menjalankan tugas *grabbing* data. Jika jumlah elemen *child* dari *document.body* dan struktur elemen masih sesuai dengan XPath yang disimpan pada Tabel 6, maka *grabbing* data akan berjalan. Jika tidak, maka aplikasi akan menghentikan kerjanya dan mengirim pemberitahuan kepada pengguna untuk melakukan update melalui menu *Setting*. Hal ini akan memicu tombol update menjadi bisa diklik.

Fitur *update* XPath akan bekerja secara otomatis dalam melakukan pembaruan ini. Aplikasi akan melakukan *scan* struktur elemen secara mandiri ke halaman web Twitter yang telah disiapkan. Pengguna tinggal menunggu beberapa saat hingga *scan* struktur elemen telah selesai dilakukan dan XPath diperbarui. Pada Tabel 8 menunjukkan XPath *Absolute* yang digunakan untuk melakukan *update* XPath sesuai halaman web Twitter yang sedang dibuka. XPath tersebut bersifat mutlak ada pada halaman yang telah disiapkan hingga didapatkan elemen yang dicari. Dari elemen tersebut aplikasi akan mengambil *parent node* hingga menemukan *tag body* pada halaman tersebut. Tersusunlah XPath baru hasil dari pengaksesan *parent node* dari elemen tersebut lalu ditampung kembali pada *local storage*.

Tabel 8 Daftar XPath Absolute

Variabel	Value	URL Halaman
xpathScanNoResults	//*[contains(text(),'No results')]	https://twitter.com/search?q=Tweet%20Grabber%20(from%3Aerindaresha)%20until%3A2020-01-01
xpathScanSearchKeyword	//input[@placeholder='Search Twitter']	
xpathScanName	(//*[text()='Erinda Resha'])[3]	
xpathScanTweetIdRaw	//*[contains(@href,'1247305302032052226')]	
xpathScanDate	//@*[contains(.,'2020-04-06T23:28:52.000Z') or contains(.,'1586215732')][1]/parent::*	
xpathScanUsername1	(//*[contains(text(), '@') and contains(text(), 'erindaresha') or contains(text(), '@') and *[contains(text(), 'erindaresha')]])[3]	
xpathScanUsername2	(//*[contains(text(), '@') and contains(text(), 'erindaresha') or contains(text(), '@') and *[contains(text(), 'erindaresha')]])[2]	
xpathScanFirstLine	(//*[.='Tweet Grabber 1'])[1]	
xpathScanSecondLine	(//*[.='Tweet Grabber 3'])[1]	

3.1.4 Fitur Pengelolaan XPath untuk Mengembalikan XPath ke Bentuk Default

Fitur mengembalikan XPath ke bentuk *default* digunakan untuk menyimpan kembali XPath pada aplikasi sesuai halaman web Twitter versi rilis 21 April 2020. Tombol ini dapat ditemukan pada menu *Setting* halaman *pop up* aplikasi di bawah tombol *update*. Fitur ini digunakan untuk mengatasi jika ada kegagalan saat melakukan *update* XPath, sehingga ketika tombol ditekan XPath yang tersimpan pada *local storage* akan kembali seperti pada Tabel 6.

3.1.5 Fitur Pengelolaan Database untuk Export Data

Fitur *export* data yang terletak pada menu *Setting* berfungsi agar pengguna dapat mengekspor data yang telah tersimpan di database menjadi *file* CSV. *File* tersebut berisi 6 kolom yang terdiri dari *tweet_id*, *created_at*, *name*, *username*, *text_content*, dan *keyword*. Contoh hasil ekspor data dapat dilihat pada Gambar 9. Dengan menggunakan dukungan dari *library* AlaSQL, mengekspor data dari IndexedDB menjadi lebih mudah hanya dengan menuliskan *query* saja.

Tweet_id	Created_at	Name	Username	Text_content	Keyword
1262725851566739456	2020-05-19 19:44:37	"YNWA" The LFC Story	@LFCHistoryShow	Sums it all up really. #Liverpoolfc	#LiverpoolFC
1263971216706150400	2020-05-23 06:13:16	Anything Liverpool	@AnythingLFC_	All smiles from the Captain #Liverpoolfc	#LiverpoolFC
1263154678071791616	2020-05-21 00:08:38	Moving Unit*	@Moving_Unit	LFC Videotape Graphic complete. Thought? Should be no	#LiverpoolFC
1262852091728531456	2020-05-20 04:06:15	Anything Liverpool	@AnythingLFC_	Mohamed Salah loves an upgrade in Fifa #Liverpoolfc #Fif	#LiverpoolFC
1264599756820680704	2020-05-24 23:50:51	Abhishek Dubey	@Abhishek_zero_7	'It was game of two halves the agony and the ecstasy. Just	#LiverpoolFC

Gambar 9. File CSV hasil dari ekspor data

3.1.6 Fitur Pengelolaan Database untuk Delete Data

Menghapus *database* dapat dilakukan dengan menggunakan fitur *delete* data. Semua rekam data yang tersimpan pada *database* akan terhapus. *Database* penyimpanan dapat diperiksa menggunakan DevTools pada halaman web Twitter, buka tab *Application* > *Storage* > *IndexedDB* > *TweetGrabberERA* > *datas*. Jika *database* telah dihapus maka *database* *TweetGrabberERA* tidak akan ditemukan.

3.2 Analisis Sistem

3.2.1 Hasil Pengujian Aplikasi

Aplikasi *TweetGrabberERA* telah yang telah selesai dari tahap pembuatan selanjutnya akan diuji untuk melihat apakah aplikasi yang dibangun telah sesuai dengan rancangan kerjanya. Hasil pengujian menunjukkan bahwa aplikasi telah bekerja sesuai rancangan kerja. Pengujian pada aplikasi berjalan seluruhnya sesuai skenario tanpa mengalami adanya kegagalan. Hasil pengujian fitur *grabbing* data yang terjadi pada aplikasi dengan kondisi menggunakan *default* XPath dan

kondisi menggunakan XPath yang telah diperbarui menunjukkan respons yang sama. Hal tersebut menandakan bahwa XPath yang telah diperbarui melalui fitur pengelolaan XPath telah berhasil. Pembaruan XPath yang terjadi tidak mengganggu jalannya fungsi kerja lain di aplikasi. Hasil yang diharapkan pada setiap skenario tes dengan hasil yang sebenarnya terjadi pada fitur *grabbing* data, fitur penjadwalan, fitur pengelolaan XPath, dan fitur pengelolaan *database* telah dinilai lulus dan sesuai.

Dapat dilihat dari pengujian ini, aplikasi mampu memberi arahan kepada pengguna untuk menjalankan aplikasi ketika pengguna tidak mengerti apa yang harus dilakukan, ketika pengguna melupakan sesuatu, atau ketika pengguna melakukan kesalahan baik sengaja maupun tidak. Selain itu aplikasi juga sudah bisa menjalankan tugas-tugasnya tanpa mengalami kendala yang berarti. Aplikasi ini berjalan secara fungsional dan tujuannya tetapi tidak disebutkan tingkat efektivitas dan efisiensinya karena penulis tidak melakukan pengujian performa dari kinerja aplikasi.

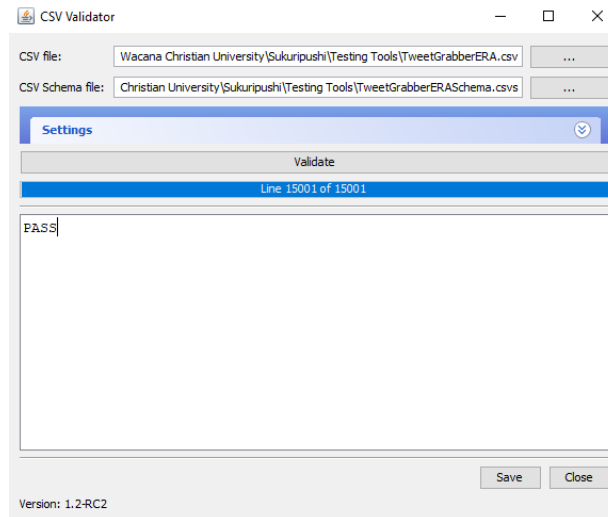
Pengujian selanjutnya yang akan dilakukan terhadap aplikasi TweetGrabberERA adalah membandingkan fitur atau fasilitas yang telah tersedia dengan aplikasi maupun program yang sama secara fungsionalitasnya. *Automatic Crawling Data in Twitter* merupakan aplikasi dari penelitian oleh Sembodo dkk. [6], memiliki salah satu fungsi yang sama dengan aplikasi TweetGrabberERA yaitu untuk melakukan pengambilan data pada Twitter. Pada Tabel 9 dapat dilihat hasil perbandingan antara TweetGrabberERA dengan *Automatic Crawling Data in Twitter*.

Tabel 9 Hasil Perbandingan TweetGrabberERA dengan *Automatic Crawling Data in Twitter*

TweetGrabberERA	<i>Automatic Crawling Data in Twitter</i>
Mengimplementasikan XPath <i>Selector</i> , tidak memerlukan melakukan pengesahan kredensial	Mengimplementasikan API Twitter, sehingga akan memerlukan <i>consumer key</i> , <i>consumer access</i> , <i>access token</i> dan <i>access secret token</i>
Metode pencarian berdasarkan <i>keyword</i>	Metode pencarian berdasarkan <i>username</i> dan <i>keyword</i>
Pada pengambilan data terakhir, <i>tweet</i> yang diambil berjumlah 15.000 dan dapat terus bertambah.	Jumlah data maksimal yang dapat diambil berdasarkan <i>username</i> adalah 200 <i>tweet</i> sedangkan jika berdasarkan <i>keyword</i> adalah 100 <i>tweet</i> .
Penyimpanan data dilakukan pada IndexedDB	Penyimpanan data dilakukan pada MySQL
Data langsung disimpan pada database, untuk melihat hasil pengambilan data harus melalui DevTools pada browser	Data diekstraksi dalam format JSON untuk ditampilkan kembali pada halaman depan aplikasi
Data dapat diekspor dengan format CSV melalui aplikasi	Untuk mengekspor data harus melalui <i>database</i>

3.2.2 Hasil Pengujian Validasi *Dataset*

Dataset hasil dari *grabbing* data akan melakukan 2 tahap pengujian. Pengujian yang pertama yaitu dengan melakukan validasi *file* format CSV berisi 15.000 rekam data yang dihasilkan oleh aplikasi TweetGrabberERA menggunakan *tool open source* CSV Validator GUI 1.2 oleh *The National Archives* dengan skema yang telah disusun pada skenario pengujian. Gambar 10 menunjukkan bahwa 15.000 rekam data pada *file* CSV telah lolos pengujian.



Gambar 10. Hasil pengujian dari CSV Validator

Selanjutnya penulis melakukan pengujian analisis sentimen dari *file* CSV menggunakan *tool open source* Orange 3.24.0. Sesuai dengan skenario yang telah dibuat, data yang akan diuji sebanyak 150 rekam *sample* data diambil secara acak dari *file* CSV yang telah tersedia melalui *tool* Orange. Metode VADER yang diterapkan untuk melakukan analisis sentimen pada *tool* menggunakan kombinasi sentimen *lexicon* yaitu suatu *library* yang terdiri dari daftar fitur leksikal seperti kumpulan kata-kata yang dilabeli menurut orientasi sematikanya [9]. Susunan alur kerja *widget* pada *tool* Orange dapat dilihat pada Gambar 11.



Gambar 1. Alur kerja *widget* pada *tool* Orange

Hasil dari metode tersebut melalui *tool* Orange akan menampilkan skor berdasarkan penilaian kata-kata yang terkumpul yang diinputkan pada kolom *positive*, *negative*, *neutral*, dan *compound* (total skor). Menurut dokumentasi Orange [10], sentimen direpresentasikan pada kolom *compound* menunjukkan angka -1 merupakan skor paling negatif, angka 1 merupakan skor paling positif. Hal tersebut membuat penulis memutuskan untuk memberi label pada rekam data menurut skor yang didapat dengan kriteria skor 0 diberi label netral, skor kurang dari 0 diberi label negatif, dan skor lebih dari 0 diberi label positif.

Hasil analisis sentimen oleh *tool* Orange yang telah diberi label selanjutnya akan dibandingkan dengan hasil analisis sentimen oleh pakar. Penulis telah mengumpulkan 5 responden yang berperan sebagai pakar untuk mengisi kuesioner analisis sentimen. Dari kelima jawaban analisis sentimen oleh pakar, akan diambil sentimen terbanyak pada tiap baris rekam data untuk digunakan sebagai perbandingan.

Pada Tabel 10 ditunjukkan *confusion matrix* hasil perbandingan prediksi sistem oleh *tool* Orange dengan hasil sebenarnya oleh pakar. Pada prediksi sistem dengan total 150 rekam data memiliki perbedaan 29 dari 70 rekam data dengan label positif yang tidak sesuai, 18 dari 64 rekam data dengan label netral yang tidak sesuai, dan 11 dari 16 rekam data dengan label negatif yang tidak sesuai. Hasil analisis sentimen oleh *tool* Orange menunjukkan bahwa label positif paling banyak terjadi sedangkan untuk hasil analisis sentimen oleh pakar menunjukkan bahwa label netral paling banyak terjadi.

Tabel 10 *Confusion Matrix*

		Hasil prediksi sistem (Orange)			Total
		Positif	Netral	Negatif	
Hasil sebenarnya (Pakar)	Positif	41	17	4	62
	Netral	28	46	7	81
	Negatif	1	1	5	7
Total		70	64	16	150

Pengujian analisis sentimen terhadap dataset file CSV yang dihasilkan oleh aplikasi TweetGrabberERA berjalan tanpa adanya halangan. Dari perbandingan antara hasil analisis sentimen oleh *tool* Orange dan pakar ditemukan 92 dari 150 rekam data memiliki hasil analisis sentimen yang sama, sehingga dengan menerapkan Rumus 2 hasil prediksi sistem dapat disebutkan memiliki nilai akurasi sebesar 61,33%.

4. KESIMPULAN

Aplikasi TweetGrabberERA yang dibangun sebagai Chrome Extension untuk data grabber media sosial Twitter telah berhasil mengimplementasikan XPath Selector dengan menerapkan method dari DOM (Document Object Model) yaitu `document.evaluate()` dan menggunakan properti `singleNodeValue`, XPath dapat diakses dan menerima hasil pengembalian dengan tipe elemen node. Walaupun menggunakan metode XPath Selector, aplikasi ini mampu beradaptasi pada struktur HTML halaman web Twitter yang berubah-ubah dengan adanya fitur *update* XPath. Cara kerja *update* XPath ini dengan memanfaatkan XPath *Absolute* pada halaman yang telah disiapkan untuk mendapatkan XPath *Relative* yang baru. XPath *Relative* inilah yang digunakan oleh aplikasi untuk mendapatkan elemen HTML dari data yang akan diambil.

Pengujian validasi *file* format CSV yang melibatkan pengecekan skema *dataset* yang berisi 15.000 rekam data berhasil dilakukan dan dinyatakan lolos. Dilakukan juga analisis sentimen melalui pemberian label sesuai skor yang dihasilkan oleh *tool* Orange dapat menunjukkan adanya sentimen positif, negatif, dan netral pada setiap rekam *sample* data yang diambil. Analisis sentimen tersebut dibuktikan validitasnya melalui perbandingan dengan analisis sentimen oleh pakar. Hasil perbandingan tersebut ditemukan bahwa persentase data yang sama antara analisis sentimen oleh *tool* Orange dengan pakar mencapai 61,33%.

5. SARAN

Pada tahap ini penulis ingin menyampaikan beberapa saran untuk dapat menunjang penelitian lebih lanjut:

1. Penelitian yang telah penulis lakukan belum membahas mengenai performa aplikasi, sehingga melakukan perbandingan aplikasi berdasarkan performa tidak dapat dilakukan. Dengan melakukan penelitian mengenai performa, diharapkan aplikasi dapat diukur tingkat efisien dan efektivitasnya.
2. Menambahkan fitur *grabbing* data berdasarkan *username* akun Twitter untuk pengembangan aplikasi selanjutnya.

DAFTAR PUSTAKA

- [1] K. Kellogg, "The 7 Biggest Social Media Sites in 2020," 3 February 2020. [Online]. Available: <https://www.searchenginejournal.com/social-media/biggest-social-media-sites/>.
- [2] Twitter, "Counting characters," tanpa tahun. [Online]. Available: <https://developer.twitter.com/en/docs/basics/counting-characters>.
- [3] Google, "What are extensions?," tanpa tahun. [Online]. Available: <https://developers.chrome.com/extensions>.
- [4] T. Rizaldi and H. Arief, "Perbandingan Metode Web Scraping Menggunakan CSS Selector dan Xpath Selector," *TEKNIKA Volume 6 Nomor 1*, 2017.
- [5] MDN Contributors, "XPath," 24 Nov 2019. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/XPath>.
- [6] J. E. Sembodo, E. B. Setiawan and Z. A. Baizal, "Data Crawling Otomatis pada Twitter," *Indonesia Symposium on Computing (IndoSC 2016)*, 2016.
- [7] Albert, K. Gunadi and E. Setyati, "Deteksi Helm pada Pengguna Sepeda Motor dengan Metode Convolutional Neural Network," 2014.
- [8] X. Deng, Q. Liu, Y. Deng and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," <http://dx.doi.org/10.1016/j.ins.2016.01.033>, 2016.
- [9] P. Pandey, "Simplifying Sentiment Analysis using VADER in Python (on Social Media Text)," 23 September 2018. [Online]. Available: <https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f>.
- [10] Orange, University of Ljubljana, "Sentiment Analysis," tanpa tahun. [Online]. Available: <https://orange.biolab.si/widget-catalog/text-mining/sentimentanalysis/>.