



Firefox Extension untuk Klasifikasi Komentar Spam pada Instagram Berbasis REST Services

Antonius Rachmat Chrismanto^{#1}, Willy Sudiarto Raharjo^{#2}, Yuan Lukito^{#3}

[#]Fakultas Teknologi Informasi, Program Studi Informatika, Universitas Kristen Duta Wacana Yogyakarta
Jl. Dr. Wahidin Sudirohusodo No. 5-25 Yogyakarta

¹anton@ti.ukdw.ac.id

²willysr@ti.ukdw.ac.id

³yuanlukito@ti.ukdw.ac.id

Abstrak— Klasifikasi komentar spam pada Instagram (IG) hanya dapat digunakan oleh pengguna melalui sistem yang berjalan di sisi client, karena data IG tidak dapat dimanipulasi dari luar IG. Dibutuhkan sistem yang dapat memanipulasi data dari sisi client dalam bentuk *browser extension*. Penelitian ini berfokus pada pengembangan *browser extension* untuk Firefox yang memanfaatkan *web services REST* pada layanan *cloud* dengan platform Amazon Web Services (AWS). *Browser extension* yang dikembangkan menggunakan 2 algoritma klasifikasi, yaitu KNN dan *Distance-Weighted KNN* (DW-KNN). *Extension* ini mampu menandai komentar spam dengan mengubah *Document Object Model* (DOM) IG menjadi berwarna merah dengan dicoret (*strikethrough*). Metode pengembangan *extension* dilakukan dengan metode *Rapid Application Development* (RAD). Pengujian pada penelitian ini dilakukan pada hasil implementasi *browser extension* dan pengukuran akurasi *web service* (algoritma KNN & DW-KNN). Pengujian implementasi *browser extension* menggunakan metode pengujian fungsionalitas, dimana setiap fitur yang telah diimplementasikan diuji apakah sudah sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Pengujian akurasi *web service* dilakukan dengan bantuan tool SOAPUI. Hasil pengujian *extension* adalah: (1) pengujian *extension* pada sembarang halaman web berhasil 100%, (2) pengujian pada halaman awal (*default*) IG berhasil 100%, (3) pengujian pada halaman profile suatu akun IG berhasil 100%, (4) pengujian pada suatu posting IG dan komentarnya, tidak selalu berhasil karena dipengaruhi oleh kemampuan algoritma pada *web services*, (5) pengujian untuk bahasa bukan Indonesia tidak selalu berhasil karena bergantung pada *library* bahasa, (6) pengujian untuk *load more comments* pada IG tidak selalu berhasil karena bergantung pada algoritma pada *web services*, dan (7) pengujian pilihan algoritma pada *options extension* berhasil 100%. Hasil akurasi rata-rata tertinggi algoritma KNN adalah 80% untuk $k=1$, sedangkan DW-KNN adalah 90% untuk $k=2$.

Kata kunci— Firefox extension, deteksi komentar spam, Instagram, KNN, DW-KNN.

I. PENDAHULUAN

Kebutuhan untuk mengurangi bahkan menghilangkan data-data yang tidak benar, tidak tepat, tidak sesuai (*hoax*), dan spam di Internet sangat tinggi seiring dengan perkembangan teknologi dewasa ini dan juga maraknya penyebaran konten yang tidak bertanggung jawab. Internet merupakan sumber data yang sangat besar dan mudah didapat, namun data yang terdapat di Internet juga memiliki kemungkinan bahwa data tersebut tidak benar, tidak sesuai, data palsu, bahkan menjadi mengganggu karena berupa data spam. Data spam banyak sekali terdapat di media sosial, di mana salah satunya terdapat di Instagram (IG). Akun-akun bot *spammer* menyerang komentar-komentar di Instagram dengan komentar-komentar spam mereka terutama terhadap publik figur. Kepopuleran IG menyebabkan banyak publik figur seperti artis dan aktor yang menggunakannya sebagai salah satu sarana promosi mereka. IG memiliki efektifitas paling tinggi dibandingkan dengan Facebook dan Twitter [1]. Dilihat dari data, Indonesia adalah pengguna terbesar Instagram se-Asia Pasifik [2]. Sayangnya, banyak sekali terdapat data berupa komentar-komentar spam pada IG publik figur Indonesia, sehingga diperlukan deteksi komentar spam yang dapat menguranginya.

Penelitian mengenai deteksi komentar spam pada IG telah banyak dilakukan. Penulis telah memulai penelitian deteksi komentar spam spesifik terhadap komentar spam pada artis dan aktor Indonesia sejak tahun 2017. Solusi yang penulis bangun dilakukan dengan menggunakan teknik deteksi komentar spam secara otomatis menggunakan sistem klasifikasi *supervised learning*. Pada tahap awal penulis melakukan implementasi metode klasifikasi dan menguji beberapa metode Support Vector Machine dan Naïve Bayes [3] sehingga terpilihlah hasil algoritma terbaik yaitu algoritma k -NN [4]. Pada tahap kedua, penulis melanjutkan dengan pengembangan *web service* berbasis REST yang mengimplementasikan metode k -NN sebelumnya. REST *service* tersebut dibangun berbasis Java Jersey yang diletakkan pada *platform*

Amazon Web Services (AWS). Pada akhir penelitian tahap kedua telah dihasilkan *web services* REST yang siap pakai [5]. Prototipe *browser extension* juga telah dibuat pada tahap kedua namun masih sangat dasar karena menggunakan GreaseMonkey, bukan *extension* yang sebenarnya [6].

Penelitian tahap kedua dianggap belum cukup, karena untuk mendeteksi, mengurangi atau bahkan menghilangkan komentar spam sampai pada level pengguna *end-user* harus dilakukan tahapan berikutnya, yaitu mengimplementasikan REST *service* yang sudah dibuat melalui *browser extension* yang mampu digunakan oleh pengguna. Harapannya, *browser extension* yang dihasilkan dapat digunakan secara langsung oleh pengguna umum. Hal ini disebabkan karena pihak *client* tidak akan dapat mengubah data IG di sisi *server*, sehingga manipulasi hanya dapat dilakukan dari sisi *client*. Perubahan hanya bisa dilakukan dari sisi tampilannya saja. Tidak hanya itu, penambahan metode klasifikasi yang lebih akurat pada *web service* juga perlu dilakukan agar pengguna dapat memilih metode klasifikasi dan dapat membandingkan hasil akurasi kedua metode tersebut.

Batasan masalah yang digunakan pada penelitian ini adalah:

1. Data pelatihan berasal dari penelitian [7].
2. Bahasa komentar IG yang digunakan adalah bahasa Indonesia.
3. *Cloud Service* yang digunakan adalah Amazon Web Service (AWS).
4. Algoritma yang digunakan pada *browser extension* adalah KNN dan *Distance Weighted* (DW) KNN.
5. *Browser extension* yang dibangun hanya digunakan pada browser Firefox.

Kontribusi penelitian ini adalah menghasilkan *browser extension* pada Firefox yang berfungsi dengan baik dan dapat digunakan secara nyata oleh pengguna umum.

II. TINJAUAN PUSTAKA

A. Penelitian Terkait

Penelitian mengenai deteksi spam pada media sosial sudah banyak dilakukan. Penelitian pertama melakukan deteksi user spam pada media sosial Twitter dan menghasilkan hasil bahwa fitur geografis, fitur persentase jumlah twit dengan *human factor index* yang tinggi, persentase twit yang mengandung URL, dan persentase twit dengan menyebutkan user lain dapat digunakan sebagai indikator penting dalam mendeteksi user spam pada Twitter. Metode yang digunakan adalah SVM, Decision Tree, Random Forest, Naïve Bayes, dan True Label. Hasil terbaik terdapat pada algoritma Random Forest. [8]. Selain pada Twitter, deteksi komentar spam juga dilakukan pada YouTube, seperti pada penelitian [9], dimana mereka menggunakan TextExpansion yang dapat menghasilkan akurasi mencapai 90.23%. Penggunaan TextExpansion yang ditambahkan metode pada regresi logistik, Naïve Bayes (Bernoulli dan Multinomial), Random Forest, dan SVM menghasilkan pengaruh yang cukup signifikan untuk meningkatkan akurasi deteksinya. Penggunaan N-Gram pada metode Random Forest, Support Vector Machine, Naïve Bayes ternyata juga membawa

pengaruh yang signifikan terhadap deteksi komentar spam pada YouTube [10]. Terakhir, penelitian [11], meneliti pengembangan *framework* untuk mendeteksi *social spam* pada berbagai media sosial dan telah dapat diimplementasikan dengan baik pada beberapa media sosial menggunakan metode Naïve Bayes.

Media sosial yang banyak dipilih orang sebagai media untuk mengekspresikan berbagai hal, misalnya aktivitas sehari-hari dan promosi adalah Instagram (IG). IG merupakan media sosial yang berfokus pada foto / gambar yang sangat populer sehingga banyak publik figur seperti aktor dan artis Indonesia menggunakannya. Sayangnya, komentar-komentar pada IG banyak sekali yang berupa spam. Deteksi komentar spam pada IG secara khusus telah dilakukan oleh banyak peneliti, seperti [12] yang telah melakukan perbandingan deteksi komentar spam pada IG menggunakan beberapa algoritma seperti Naïve Bayes, SVM, dan XGBoost yang diberi library FastText untuk *feature set*-nya. Ternyata penggunaan FastText dapat menghasilkan F1 *score* 0.9523. Penelitian [13] juga melakukan deteksi komentar spam pada IG dan menghasilkan akurasi 96.27% menggunakan beberapa metode *supervised learning* untuk klasifikasi.

Penelitian deteksi komentar spam pada IG yang telah disebutkan di atas memang telah banyak dilakukan, namun berdasarkan data-data yang ada, penelitian-penelitian tersebut terbatas hanya pada pengujian akurasi metode-metode tertentu saja. Padahal kebutuhan deteksi spam otomatis diperlukan oleh pengguna umum secara nyata agar dapat digunakan secara langsung. Kesulitan yang dihadapi para peneliti adalah data IG yang tidak bisa diubah (*read-only*) oleh *client* karena adanya aturan tentang Web Origin [14] [15]. Data IG terdapat pada sisi *server*, sedangkan untuk menghasilkan aplikasi deteksi komentar spam otomatis adalah harus mengubah data IG secara langsung, yang pasti tidak akan bisa dilakukan, kecuali IG sendiri yang membuat aplikasinya. Oleh karena itu penulis telah melakukan beberapa tahapan penelitian sejak tahun 2017 guna mempersiapkan implementasi deteksi komentar spam otomatis yang bisa langsung diterapkan oleh pengguna.

Pada penelitian pertama (2017), telah dilakukan tahapan pengambilan data status dan komentar dari IG sepuluh artis Indonesia yang ber-*follower* paling banyak (lebih dari 10.000.000 *follower*), sehingga telah diperoleh data *training* untuk digunakan dalam metode *supervised learning*. Pada penelitian tersebut telah diteliti 3 metode yang berbeda, yaitu metode Naïve Bayes, dengan keakuratan tertinggi mencapai 77.5% [7], metode Support Vector Machine (SVM), dengan keakuratan tertinggi mencapai 78.5% [3], dan metode k-NN dengan keakuratan tertinggi mencapai 88.4% [4]. Dari hasil-hasil tersebut, diperoleh metode terbaik saat ini adalah metode k-NN, sehingga pada penelitian tahun 2018 dilanjutkan dengan melakukan pengembangan layanan deteksi komentar spam berbasis *web services* pada Amazon Web Services (AWS). Berbasis REST *service* yang telah dikembangkan [5], maka penelitian ini akan menggunakan layanan yang ada untuk dilanjutkan dengan pengembangan *browser extension* yang bukan hanya sekedar prototipe [6] sehingga dapat langsung

digunakan oleh pengguna IG secara umum untuk mendeteksi komentar spam berbahasa Indonesia.

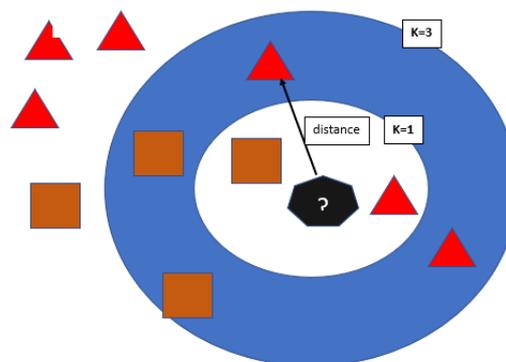
Berdasarkan statistik *browser market share* Desember 2018, posisi teratas diduduki oleh Chrome, Safari, Edge, dan Firefox [16]. Pengembangan *web browser extension* akan difokuskan pada browser Firefox karena memiliki struktur pengembangan *extension* yang mirip satu sama lain [17].

Browser extension memang tidak begitu banyak dipublikasikan dalam penelitian, namun terdapat beberapa penelitian yang menggunakan *browser extension* untuk membantu standarisasi pengambilan data untuk biomedis seperti CEDAR untuk memasukan data *ontology* [18]. Berbeda dengan penelitian sebelumnya, Privacy Meter merupakan sebuah *browser extension* yang digunakan untuk pengambilan data secara *crowdsourced* berkaitan dengan seberapa tingkat kesadaran seseorang terhadap privacy melalui browser saat ber-Internet [19]. Terakhir, browser extension juga digunakan pada penderita disleksia dalam membaca di browser [20]. Semua *browser extension* tersebut digunakan untuk meningkatkan kegunaan browser untuk membantu penggunaanya.

Selain pengembangan *browser extension*, penelitian ini juga berusaha memberikan alternatif metode klasifikasi dan memperbaiki akurasi metode deteksi klasifikasi komentar spam yang sampai saat ini masih menggunakan k-NN. Salah satu metode yang merupakan pengembangan dari metode k-NN dengan akurasi yang lebih baik adalah *Distance Weighted K-NN* [21], yang kemudian memiliki banyak varian seperti pada penelitian [22], [23], [24]. Metode DW-KNN ini mampu menjadi metode yang lebih baik daripada k-NN karena memiliki bobot tertentu menggunakan rumus khusus yang hasil jaraknya diolah kembali untuk mendapatkan pembobotan tiap jarak, yang nantinya dipakai dalam proses klasifikasi dari data itu sendiri, sehingga pengambilan keputusan dari klasifikasi suatu data yang baru tidak hanya sekedar dari jumlah kelas terbanyak dari jumlah k yang ada, namun dihitung berdasarkan rumus dengan pembobotan yang baru untuk menghindari sensitivitas dari k-NN itu sendiri dalam melakukan klasifikasi.

B. K-Nearest Neighbor

K-Nearest Neighbor (K-NN) merupakan metode yang menggunakan nilai statistik dalam mencari tetangga terdekat dari suatu pencarian data. Algoritma K-NN bekerja dengan mencari sejumlah k obyek data atau pola yang terdekat dengan data input, kemudian dilanjutkan dengan memilih kelas yang memiliki jumlah pola terbanyak di antara k pola tersebut. K-NN menggunakan prinsip *voting* dalam mengklasifikasikan pola seperti pada Gambar 1.



Gambar 1. Ilustrasi algoritma k-NN

Penentuan k pola terdekat didasarkan aturan jarak, *similarity* atau *dissimilarity*, berdasarkan pada macam atributnya. Berikut cara kerja algoritma k-NN: [25]:

- a) Untuk setiap pola latih $\langle x, f(x) \rangle$, tambahkan pola tersebut ke dalam daftar data latih
- b) Untuk sebuah pola masukan x_i :
 - i. Misakan x_1, x_2, \dots, x_k adalah k pola yang memiliki jarak terdekat (tetangga) dengan x_i . Dimana jarak terdekat dihitung menggunakan *Euclidian Distance* [26]
 - ii. Kembalikan kelas yang memiliki jumlah pola paling banyak di antara k pola tersebut sebagai kelas keputusan.

Berdasarkan penjelasan sebelumnya, metode k-NN memiliki kelebihan [25]:

1. Mudah dibuat dan mudah diimplementasikan
2. K-NN bekerja lokal dengan memperhitungkan sejumlah k data sehingga tepat dengan himpunan data yang terkelompok secara lokal (dalam sebuah kelas terdapat sejumlah kumpulan data yang saling terpisah)

Metode k-NN juga memiliki kekurangan [27]:

1. Memiliki waktu yang tidak efisien, karena selalu mengulang perbandingan, tidak menyimpan informasinya. Tidak efisien untuk data yang banyak.
2. Masalah nilai k yang tidak bisa ditentukan dengan rumus matematika, sehingga diperlukan observasi terhadap nilai k yang terbaik
3. Penentuan nilai k genap juga akan membuat sistem menjadi tidak bisa menentukan keputusan yang tepat. Nilai k yang rendah akan menyebabkan generalisasi, sedangkan nilai k besar akan menyebabkan generalisasi rendah (*overfitting*).

Karena k-NN merupakan algoritma terbaik berdasarkan penelitian pada tahap pertama maka algoritma k-NN akan digunakan dalam penelitian ini sebagai algoritma utama yang diterapkan pada *web service* deteksi spam pada IG yang dibangun.

C. Distance Weighted k-NN

DW-KNN merupakan pengembangan pada K-Nearest Neighbor yang sudah ada dengan menggunakan tambahan pembobotan (*dual weight*) dari nilai k yang sudah

ditentukan. Dalam perkembangannya, metode k-NN diperbaiki dengan tambahan pembobotan (*dual weight*). Tujuannya adalah untuk mengurangi efek sensitivitas yang dimiliki oleh k-NN dengan nilai k yang kecil, mengurangi sensitivitas jika jumlah data suatu kelas terlalu besar, atau jika nilai k yang terlalu besar dan beberapa hal lainnya [23]. Penggunaan DW-kNN dan rumus *dual weight* ini akan membuat pembobotan dengan nilai 1 untuk dokumen dengan jarak yang terdekat, nilai 0 untuk dokumen dengan jarak yang terjauh, dan nilai di antaranya untuk data yang berada diantara yang terbesar dan terkecil. Pembobotan lanjutan yang digunakan pada DW-kNN ini memiliki rumus yang dapat dilihat pada Gambar 2, dimana k merupakan data terjauh dari data uji, dan i merupakan angka perulangan dari hasil jarak yang sudah diurutkan sebelumnya secara *ascending*.

$$\bar{w}_i = \begin{cases} \frac{d(\bar{x}, x_k^{NN}) - d(\bar{x}, x_i^{NN})}{d(\bar{x}, x_k^{NN}) - d(\bar{x}, x_1^{NN})} \times \frac{d(\bar{x}, x_i^{NN}) + d(\bar{x}, x_1^{NN})}{d(\bar{x}, x_k^{NN}) + d(\bar{x}, x_1^{NN})}, & \text{if } d(\bar{x}, x_k^{NN}) \neq d(\bar{x}, x_1^{NN}), \\ \frac{d(\bar{x}, x_k^{NN}) - d(\bar{x}, x_i^{NN})}{d(\bar{x}, x_k^{NN}) - d(\bar{x}, x_1^{NN})} \times \frac{d(\bar{x}, x_i^{NN}) + d(\bar{x}, x_1^{NN})}{d(\bar{x}, x_k^{NN}) + d(\bar{x}, x_1^{NN})}, & \text{if } d(\bar{x}, x_k^{NN}) = d(\bar{x}, x_1^{NN}). \end{cases}$$

Gambar 2. Rumus bobot DWKNN [20]

Langkah pengerjaan yang membedakan DW-kNN dengan k-NN ini sendiri pada pada perhitungan bobot tambahan (*dual weight*), yang diambil dari perhitungan jarak k data terdekat seperti tahap terakhir pada pengerjaan kNN. Bobot tambahan ini nantinya akan digunakan dalam klasifikasi kelas dengan melakukan penggabungan dari bobot tambahan dan juga jumlah kelas yang ada pada k data tersebut.

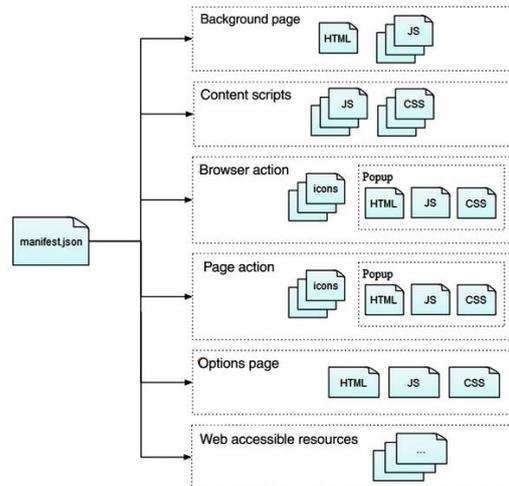
D. Browser Extension (Plugin)

Browser extension adalah aplikasi tambahan yang diinstall pada browser yang mampu menambah kemampuan dan fungsi standar dari sebuah *browser*. *Browser extension* memiliki kelebihan dan kemampuan seperti web API seperti menggunakan JavaScript pada sebuah halaman web, namun *browser extension* juga memiliki akses tersendiri terhadap kumpulan API JavaScript.

Browser Extension untuk Firefox dibangun menggunakan WebExtensions API, yang bersifat *cross-browser system*. Beberapa API yang disediakan bahkan kompatibel dengan *extension* API yang didukung oleh Google Chrome dan Opera. Extension yang dibuat sebagian besar akan jalan dengan baik di Firefox atau Microsoft Edge dengan sedikit perubahan. API nya juga akan dapat mengakses dan menggunakan kemampuan *multiprocessing* Firefox [28].



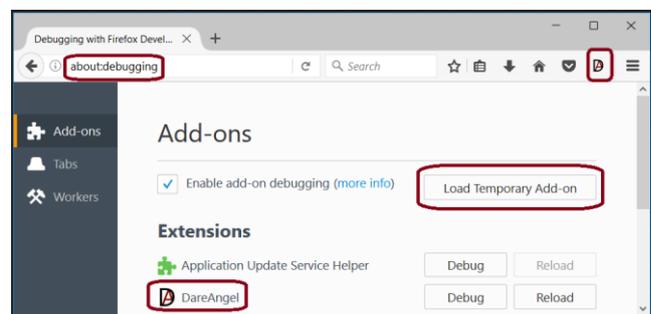
Gambar 3. Arsitektur browser extension [17]



Gambar 4. Struktur manifest [28]

Arsitektur dari *browser extension* adalah terdiri dari: manifest dalam bentuk JSON, konten dalam bentuk JavaScript, dan user interface dalam bentuk HTML. Gambar 3 menjelaskan arsitektur *browser extension*.

Pada penelitian ini browser yang akan dibuat extensionnya adalah Firefox. Struktur manifest untuk Firefox dapat dilihat di halaman *developer* Mozilla¹. Manifest sendiri terdiri dari beberapa bagian yaitu *background page*, *content script*, *browser action*, *page action*, *option page*, dan *web resources*. Struktur manifest dapat dilihat pada Gambar 4. Gambar *Firefox extension* dapat dilihat pada Gambar 5.



Gambar 5. Firefox extension [17]

¹ <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/manifest.json>

III. METODOLOGI PENELITIAN

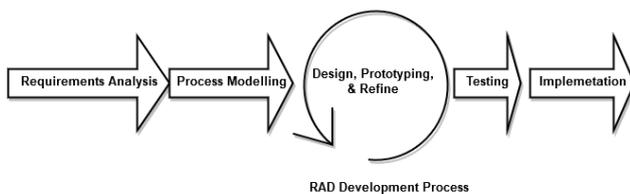
A. Tahapan Penelitian

Tahapan penelitian yang dilakukan pada penelitian ini adalah sebagai berikut :

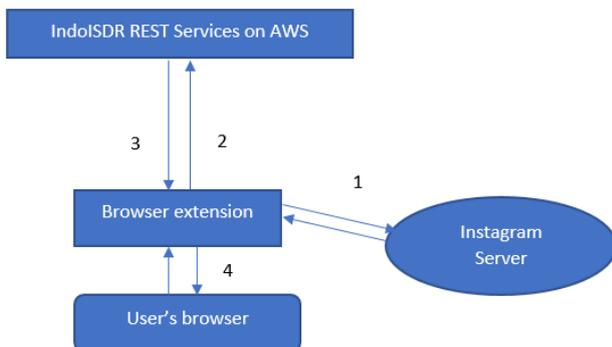
1. Tahap studi literatur, digunakan untuk mempelajari literatur tentang integrasi web services, pengembangan *browser extension* pada Firefox, dan metode DW-kNN.
2. Tahap implementasi metode DW-kNN pada web services REST.
3. Tahap implementasi *browser extension* yang menggunakan *services* dari REST services pada penelitian sebelumnya pada browser Firefox. Pada tahap ini digunakan metode pengembangan software *Rapid Application Development* (RAD) yang tahapannya dapat dilihat pada Gambar 6.
4. Tahap pengujian dan evaluasi, dimana pada tahap ini dilakukan pengujian terhadap akurasi metode k-NN yang telah digunakan sebelumnya dengan metode DW-KNN. Pengujian juga dilakukan terhadap *browser extension* yang dibuat agar sesuai dengan fungsinya.

B. Arsitektur Sistem

Browser extension dipilih sebagai salah satu cara mengimplementasikan aplikasi deteksi komentar spam otomatis karena melalui *browser extension client* dapat mengubah data pada “tampilan” IG, bukan data aslinya di server, sehingga melalui “tampilan” dari sisi client tersebut sistem dapat memanipulasi tampilan sehingga deteksi komentar spam dapat diterapkan. Model arsitektur sistem yang akan dibuat dapat dilihat pada Gambar 7.



Gambar 6. Tahapan pengembangan *Rapid Application Development* (RAD)



Gambar 7. Arsitektur sistem

TABEL I
DAFTAR KEBUTUHAN FUNGSIONAL

No.	Kebutuhan Fungsional	Modul / Solusi	Keterangan
1	Deteksi URL	Regex pada URL browser	Javascript
2	Parsing div komentar	Regex DOM div komentar pada Instagram	Javascript
3	Deteksi bahasa	Library guessLanguage based on NLP	URL: https://github.com/richtr/guessLanguage.js/
4	Deteksi komentar spam	Request ke REST services	Ada 2 services: http://tomcat.zdbgm.wikga.us-east-2.elasticbeanstalk.com/ https://vhost.ti.ukdw.ac.id/~sastrawi/
5	Koneksi ke REST service	XMLHTTP Request	Permission pada extension
6	Pemilihan algoritma	Options pada browser	Promise dan permission Storage Javascript
7	Pengubahan DOM komentar	Regex DOM dan implementasi CSS	Javascript
8	Mendeteksi perubahan data komentar pada Instagram (load more)	Library arrive.js	URL: https://github.com/uzairfarooq/arrive

Dari gambar 7 dapat dijelaskan alur kerja dan arsitektur sistem sebagai berikut:

1. User membuka *browser* dan mengirimkan HTTP *request* ke alamat IG yang mengandung post dan komentar yang kemungkinan mengandung komentar spam.
2. *Browser extension* yang telah terpasang pada *browser* pengguna akan mem-parsing data HTML dari halaman IG yang ditampilkan browser terutama pada bagian komentar-komentar. Semua bagian komentar akan dikirim (*request*) ke IndoISDR REST *service* yang telah dibangun pada penelitian tahun kedua.
3. IndoISDR akan mendeteksi komentar spam dan mengembalikan hasil deteksi dalam bentuk format JSON kepada *browser extension*.

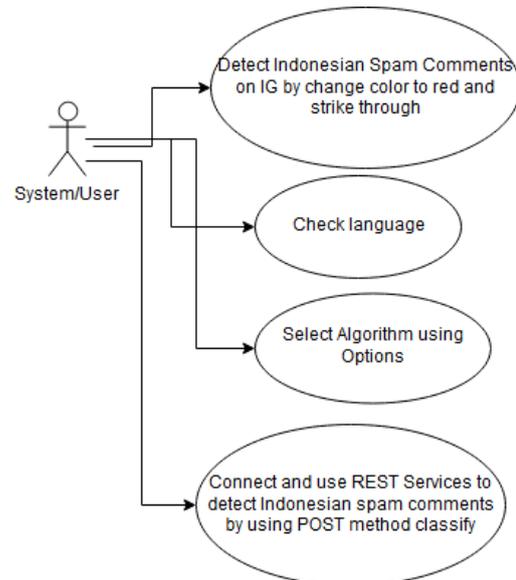
Berdasarkan deteksi dari IndoISDR REST service maka *browser extension* akan memanipulasi dokumen HTML IG pada sisi browser client untuk diubah tampilannya dan semua komentar yang berupa spam akan diberi tanda tertentu sehingga pengguna umum dapat melihat langsung deteksinya. Semua komentar spam juga dapat dihilangkan dari sisi client. Karena perubahan terjadi hanya pada sisi client secara lokal dan tidak mengakses

object yang dianggap sensitif, maka hal ini tidak melanggar aturan *Same Origin Policy* [29] [30].

C. Perancangan Kebutuhan Fungsional

Kebutuhan fungsional dari sistem *browser extension* ini adalah sebagai berikut:

1. Kebutuhan untuk mendeteksi URL instagram
 2. Kebutuhan parsing tempat komentar pada instagram
 3. Kebutuhan untuk mendeteksi bahasa yang digunakan
 4. Kebutuhan mendeteksi komentar spam menggunakan *REST Services*
 5. Kebutuhan koneksi ke *REST Service*
 6. Kebutuhan memilih penggunaan algoritma *REST Service*
 7. Kebutuhan mengubah DOM pada elemen komentar menjadi *strikethrough* berwarna merah
 8. Kebutuhan menampilkan dan mendeteksi perubahan / penambahan data komentar secara otomatis
- Detail kebutuhan dan solusinya dapat dilihat pada Tabel I.

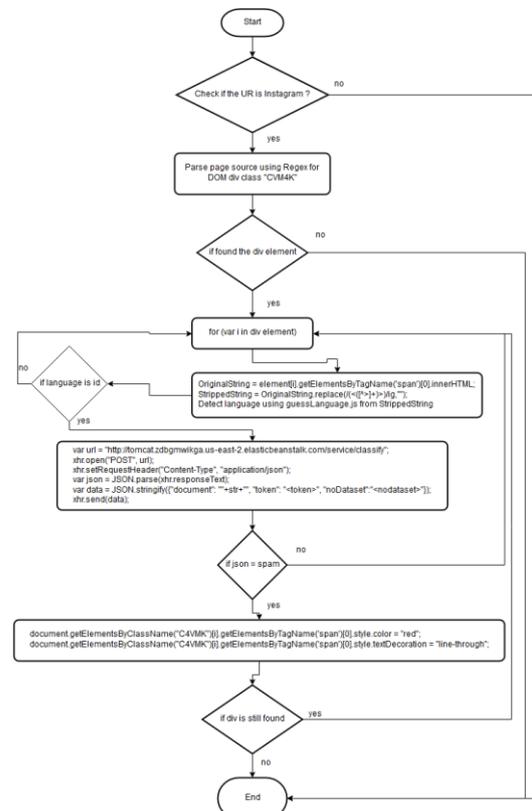


Gambar 8. Use Case Diagram

D. Use Case dan Flowchart

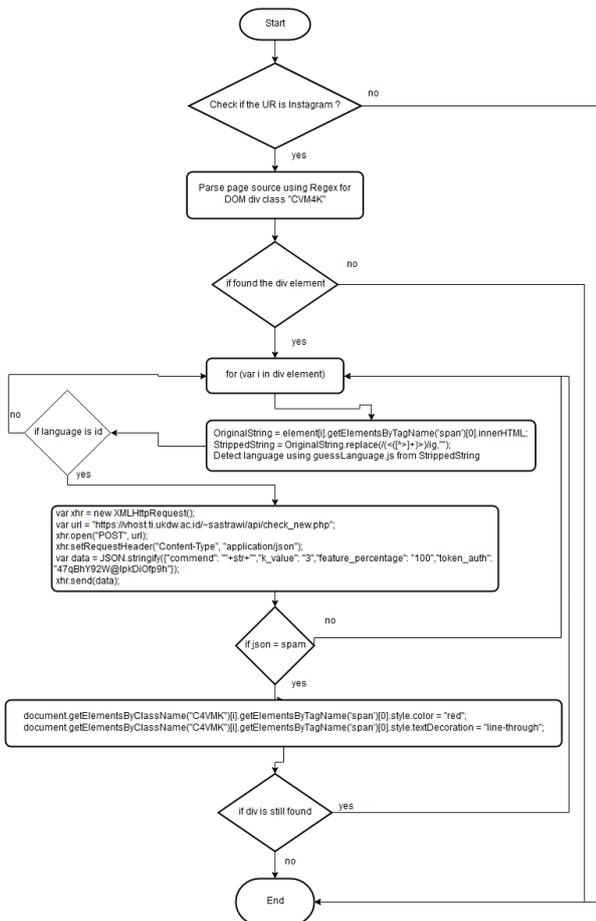
Proses bisnis pada sistem yang dibuat adalah sebagai berikut:

1. Sistem mendeteksi URL pada *browser* pengguna
2. Jika ternyata mengandung alamat URL <https://www.instagram.com/something/> maka sistem akan mulai memarsing ada atau tidaknya tag `<div>` pada kode halaman instagram yang berisi letak status dan komentar
3. Untuk setiap tag `<div>` tersebut sistem akan mengambil isi teks dari komentar dan melakukan *pre-processing*, yaitu pembersihan data
4. Sistem akan mengirimkan hasil *pre-processing* ke bagian deteksi bahasa
5. Jika dikenali sebagai bahasa Indonesia maka sistem akan melanjutkan proses deteksi komentar spam dengan menggunakan web services REST yang telah dibuat pada penelitian sebelumnya [5]
6. *REST service* dipilih sesuai pengguna apakah memilih algoritma KNN atau DW-KNN sesuai pilihan pada *input option* pada setting *browser extension*
7. Jika pengguna memilih algoritma KNN akan menggunakan service IndoISDR pada alamat URL <http://tomcat.zdbgmwikga.us-east-2.elasticbeanstalk.com/>, sedangkan jika dipilih algoritma DW-KNN akan menggunakan service pada alamat URL <https://vhost.ti.ukdw.ac.id/~sastrawi/>
8. Layanan *web service* akan memproses data dari *browser extension* dan mengembalikan hasilnya.
9. Hasil kembalian dari *web service* diperiksa. Jika berupa “spam” maka tag `<div>` yang berisi komentar akan dimodifikasi menjadi berwarna merah dan dicoret. Jika kembalian berupa “notspam” maka tidak diubah sama sekali.



Gambar 9. Flowchart pilihan algoritma KNN

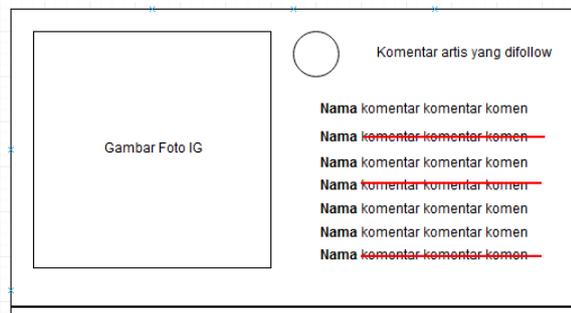
Use case sistem dapat dilihat pada Gambar 8, sedangkan *flowchart* kerja pemilihan web service REST untuk metode KNN dan DW-KNN dapat dilihat pada Gambar 9 dan 10.



Gambar 10. Flowchart pilihan algoritma DW KNN

E. Perancangan User Interface

Browser extension yang dibuat tidak memiliki user interface karena cara kerjanya langsung tanpa harus menunggu input apapun dari pengguna. Saat halaman URL berasal dari Instagram, maka plugin akan langsung bekerja dan memarsing semua komentar dan mendeteksi spam atau bukan. Desain tampilan hasil yang diharapkan dapat dilihat pada Gambar 11.



Gambar 11. Desain Hasil Browser Extension pada Instagram

F. Perancangan Pengujian

Pada tahap ini akan dilakukan pengujian terhadap browser extension pada Firefox dengan skenario test sebagai berikut:

1. Pengujian sembarang halaman web
2. Pengujian halaman awal (default) Instagram
3. Pengujian halaman profil suatu akun pada Instagram

4. Pengujian suatu posting akun Instagram dan komentarnya
5. Pengujian untuk bahasa non Indonesia
6. Pengujian untuk load more comments
7. Pengujian pilihan algoritma pada web service
8. Pengujian terhadap akurasi metode kNN dan DW-kNN.

IV. HASIL DAN PENGUJIAN

A. Implementasi Struktur Folder

Setelah dilakukan perancangan seperti pada bab III sebelumnya, implementasi browser extension dilakukan dengan pengelolaan struktur direktori sebagai berikut:

```

icons\
lib\
about.html
arrive.js
ig.js
manifest.json
options.html
options.js
package.json
README.md
    
```

Folder icons digunakan untuk menyimpan icon aplikasi, lib adalah folder opsional untuk pustaka javascript pihak ketiga, about.html adalah file untuk menampilkan informasi mengenai aplikasi pada toolbar Firefox, arrive.js adalah file pihak ketiga untuk javascript, ig.js adalah file javascript utama aplikasi, manifest.js adalah informasi meta data tentang aplikasi dan setting aplikasi, options.html adalah file untuk setting pada browser extension, options.js adalah file javascript yang digunakan oleh options.js, package.json adalah setting package, dan README.md adalah file readme.

B. Implementasi Manifest.JSON

File manifest.json adalah file utama yang berisi informasi meta data pada browser extension pada Firefox. File ini adalah file berekstensi dan berformat JSON yang berisi beberapa informasi yaitu:

- Versi Manifest. Bagian ini berisi versi file manifest, yaitu 2 sesuai dengan tutorial pada Firefox Extension Developer Web.
- Nama. Bagian ini berisi nama browser extension.
- Versi Aplikasi. Bagian ini berisi versi browser extension.
- Icon. Bagian ini berisi referensi file icon yang digunakan pada browser extension.
- Browser action. Bagian ini berisi option pada toolbar browser, seperti menampilkan about aplikasi.
- Content script. Bagian ini berisi script javascript untuk program utama.
- Options_UI. Bagian ini berisi setting file html untuk user interface setting browser.
- Permission. Bagian ini berisi permission yang dibutuhkan oleh aplikasi, seperti kebutuhan akses penyimpanan, kebutuhan akses alamat URL lain untuk mencegah terjadinya cross browser policy.

- *Application*. Bagian ini berisi identitas email pembuat aplikasi dan ID nya.

C. Implementasi Package.json

Package.json adalah file yang dibutuhkan mengenai informasi packaging *browser extension* yang dibuat. Beberapa bagian JSON yang harus ada adalah:

- *Title*, untuk informasi judul package.
- *Name*, untuk informasi nama package.
- *Versi*, untuk informasi versi package.
- *Description*, untuk informasi deskripsi package.
- *Main*, yaitu berisi file javascript utama aplikasi.
- *Author*, yaitu berisi nama pembuat aplikasi.
- *Engines*, yaitu berisi informasi versi Firefox yang digunakan.
- *License*, yaitu informasi lisensi yang digunakan
- *Keywords*, yaitu informasi kata kunci pencarian aplikasi.

D. Implementasi Javascript

Implementasi *ig.js* dibuat berdasarkan proses bisnis dan *flowchart* pada sub bab sebelumnya. *ig.js* merupakan file program utama aplikasi yang dibangun dengan kode sebagai berikut.

1. Fungsi utama

Fungsi tersebut pertama melakukan pengecekan URL Instagram, kemudian setelah sesuai, maka dicari elemen *div class="C4VMK"* untuk diparsing dan diambil informasi teks komentarnya. Teks komentarnya diparsing dan dibersihkan semua tag HTMLnya serta dideteksi bahasanya, apakah bahasa Indonesia atau bukan. Jika bahasa Indonesia maka dilanjutkan dengan pemilihan algoritma, apakah kNN atau DW-kNN tergantung sistem algoritma yang dipilih oleh pengguna berdasarkan preferensinya.

2. Fungsi uji klasifikasi

Fungsi ini menggunakan web service REST dari untuk metode kNN dan DW-kNN. Methodnya menggunakan POST dan data yang dikirimkan menggunakan format JSON. Jika sudah selesai melakukan deteksi, maka dilanjutkan dengan mengubah komentar menjadi berwarna merah dan dicoret.

3. Fungsi simpan *options* algoritma

Fungsi ini menggunakan Javascript untuk menyimpan dan mengambil data dari storage browser berdasarkan pilihan pengguna dari *option extension browser*.

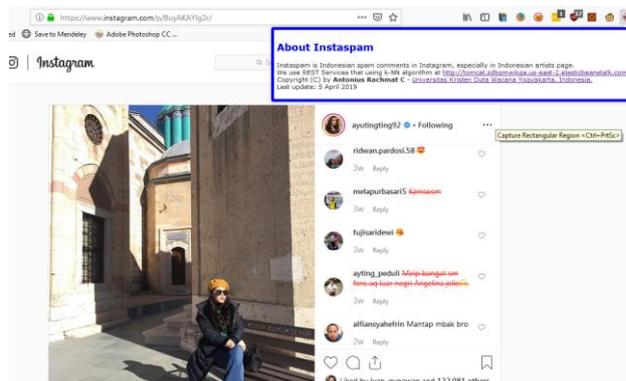
E. Implementasi Options, About, dan Library lain

Skrip *options* berupa HTML yang memanggil skrip javascript untuk menyimpan data hasil preferensi pengguna mengenai algoritmanya, sedangkan skrip *about* berupa file HTML untuk menampilkan informasi tentang aplikasi dan pembuatnya.

File library lain yang digunakan adalah file *arrive.js* dan library *guessLanguage.js*. *Arrive.js* adalah library yang

dibuat untuk mendeteksi suatu komponen DOM berubah (dihapus / ditambah)². *GuessLanguage.js* adalah library untuk mendeteksi bahasa yang diinputkan oleh pengguna berbasis NLP³.

Hasil akhir dari *browser extension* yang telah dibuat dapat dilihat pada gambar 12 dan 13.



Gambar 12. Hasil Browser Extension



Gambar 12. Hasil Deteksi Komentar Spam

F. Hasil Pengujian Fungsionalitas Sistem

Pada tahap ini dilakukan testing browser extension pada Firefox dengan skenario seperti pada bab 3 sebelumnya dengan hasil dapat dilihat pada Tabel II.

² <https://github.com/uzairfarooq/arrive>

³ <https://github.com/richtr/guessLanguage.js>

TABEL II
HASIL PENGUJIAN SISTEM

N o	Pengujian	Hasil Sistem	Hasil yang diharapkan	Keterangan
1.	Pengujian pada sembarang halaman web	Berhasil / Tepat	Berhasil / Tepat	Semua halaman web selain IG tidak akan dideteksi oleh browser plugin.
2.	Pengujian pada halaman awal (default) Instagram	Berhasil / Tepat	Berhasil / Tepat	Tampilan pada halaman https://www.instagram.com tidak dideteksi komentar spam
3.	Pengujian pada halaman profile suatu akun	Berhasil / Tepat	Berhasil / Tepat	Pada halaman profile depan IG hanya berisi image / foto sehingga tidak ada deteksi komentar spam
4.	Pengujian pada suatu posting akun Instagram dan komentarnya	Berhasil / Tepat	Berhasil / Tepat	Pada halaman suatu posting akun IG, yang berbahasa Indonesia akan dideteksi komentar spamnya. Namun akurasi benar/salah deteksinya sangat tergantung pada web service yang digunakan
5.	Pengujian untuk bahasa non Indonesia	Berhasil namun tidak 100%	Berhasil / Tepat	Pada halaman posting akun IG yang tidak berbahasa Indonesia kebanyakan tidak dideteksi, namun beberapa komentar dianggap bahasa Indonesia. Hal ini bergantung pada hasil dari pustaka guessLanguage.js yang berbasis NLP.
6.	Pengujian untuk load more comments	Berhasil namun tidak 100%	Berhasil / Tepat	Pada halaman load more comments, sistem sudah bisa dan tetap mendeteksi komentar spam, namun beberapa komentar spam tidak bisa dideteksi dengan baik. Hal ini karena faktor kemampuan algoritma web services
7.	Pengujian pilihan algoritma pada web service	Berhasil, namun tidak bisa dari awal	Berhasil / Tepat	Bagian options pada extension bisa berfungsi dengan baik, namun browser extension hanya bisa memilih setelah load more comments.

G. Hasil Pengujian Akurasi KNN dan DW-KNN

Pengujian akurasi pada penelitian ini dilakukan dalam beberapa skenario pengujian dengan beberapa parameter yang diubah-ubah, antara lain nilai k (untuk k=1, 2 dan 3) pada metode KNN dan DW-KNN, stemming (menggunakan stemming atau tidak) dan *features selection* (dari 50% sampai 100%). Hasil pengujian untuk parameter k=1, 2 dan 3 dapat dilihat pada Tabel III-V.

Dari Tabel III-V dapat dilihat bahwa pemilihan nilai k dapat mempengaruhi tingkat akurasi yang didapatkan. Jika menggunakan nilai k=1, hasil akurasi antara metode DW-KNN dan KNN sama baiknya. Hal ini disebabkan pada saat parameter k=1, maka metode DW-KNN sebenarnya sama saja dengan metode KNN dimana tidak ada pembobotan jarak data uji dengan data acuan.

Pada nilai k=2 dan k=3 didapatkan tingkat akurasi DW-KNN lebih baik daripada KNN. Hal ini disebabkan karena pada metode DW-KNN dilakukan pembobotan dari setiap jarak data uji dengan data acuan, sehingga hasilnya akan lebih baik daripada metode KNN yang tidak menggunakan pembobotan. Perbedaan jumlah fitur (*features selection*) yang dipakai untuk klasifikasi ternyata tidak ada pengaruhnya. Pada saat nilai k=3 terlihat ada kenaikan akurasi jika jumlah fitur yang digunakan semakin sedikit, namun tingkat kenaikan akurasi secara keseluruhan dapat dikatakan tidak banyak berbeda atau tidak signifikan. Gambar 13 merupakan grafik yang menunjukkan tingkat akurasi pengujian yang dilakukan.

Parameter *features selection* yang semakin mengecil dapat meningkatkan akurasi karena pada tahap tersebut dilakukan pengurutan mulai dari fitur-fitur yang dianggap paling berpengaruh dan menghilangkan fitur-fitur yang dianggap kurang berpengaruh. Jika ingin menggunakan *features selection* disarankan dilakukan pada saat nilai k diperbesar (mulai dari k=3, 4, dan seterusnya). Pada penelitian ini penggunaan tahap *stemming* ternyata tidak banyak memberikan perbedaan akurasi karena data komentar pada IG biasanya berupa kalimat pendek yang tidak banyak kata-kata berulang, sehingga tahap *stemming* tidak banyak memberikan kontribusi terhadap akurasi.

TABEL III
HASIL AKURASI K=1 (STEMMING DAN NON STEMMING)

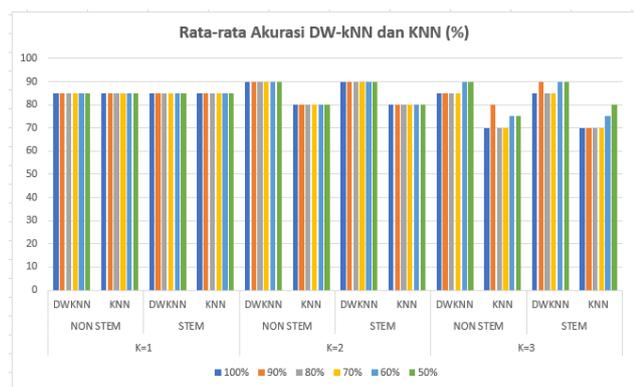
% Features Selection	K=1			
	NON STEM		STEM	
	DW-KNN	K-NN	DW-KNN	K-NN
100	85	85	85	85
90	85	85	85	85
80	85	85	85	85
70	85	85	85	85
60	85	85	85	85
50	85	85	85	85
Rata	85	85	85	85

TABEL IV
HASIL AKURASI K=2 (STEMMING DAN NON STEMMING)

% Features Selection	K=2			
	NON STEM		STEM	
	DW-KNN	K-NN	DW-KNN	K-NN
100	90	80	90	80
90	90	80	90	80
80	90	80	90	80
70	90	80	90	80
60	90	80	90	80
50	90	80	90	80
Rata	90	80	90	80

TABEL V
HASIL AKURASI K=3 (STEMMING DAN NON STEMMING)

% Features Selection	K=3			
	NON STEM		STEM	
	DW-KNN	K-NN	DW-KNN	K-NN
100	85	70	85	70
90	85	80	90	70
80	85	70	85	70
70	85	70	85	70
60	90	75	90	75
50	90	75	90	80
Rata	86.66667	73.3333	87.5	72.5



Gambar 13. Rata-rata akurasi

V. KESIMPULAN

Dari penelitian yang telah dilakukan, maka didapat beberapa kesimpulan sebagai berikut:

1. Implementasi *browser extension* telah berhasil dilakukan dan menghasilkan sebuah produk berupa *browser extension* pada browser Firefox yang dapat memanfaatkan *web services* berbasis REST dan melakukan deteksi komentar spam pada layanan media sosial Instagram.
2. Dari hasil pengujian fungsional sistem, semua fungsi berhasil dibuat dan berjalan dengan baik, namun tidak semuanya berjalan sempurna dengan tepat 100%, yaitu pada bagian deteksi bahasa dan deteksi komentar spam. Hal ini terjadi karena ketergantungan antara *browser extension* dengan kemampuan dari library deteksi spam dan algoritma K-NN & DW-kNN yang menjadi pilihan pengguna dalam menggunakan *browser extension* tersebut.
3. Berdasarkan pengujian akurasi, algoritma K-NN memiliki rata-rata akurasi tertinggi 85% untuk k=1 dan DW-KNN memiliki rata-rata akurasi tertinggi 90% untuk k=2.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) Universitas Kristen Duta Wacana Yogyakarta (UKDW) atas bantuan dana dan infrastruktur sehingga penelitian dengan surat kontrak No. 069/D.01/LPPM/2019 ini dapat berjalan dengan baik dan lancar.

REFERENSI

- [1] Darmawan, "Maxmanroe.com," 2017. [Online]. Available: <https://www.maxmanroe.com/belajar-instagram-marketing-dari-hasil-analisa-terhadap-26-akun-instagram-artis-indonesia.html>.
- [2] F. K. Bohang, "Indonesia, Pengguna Instagram Terbesar se-Asia Pasifik," 27 07 2017. [Online]. Available: <https://tekno.kompas.com/read/2017/07/27/11480087/indonesia-pengguna-instagram-terbesar-se-asia-pasifik>.
- [3] A. R. Chrismanto dan Y. Lukito, "Identifikasi Komentar Spam Pada Instagram," Lontar Komputer, vol. 8, no. 3, pp. 219-231, 2017.
- [4] A. R. Chrismanto dan Y. Lukito, "Klasifikasi Komentar Spam Pada Instagram Berbahasa Indonesia Menggunakan K-NN," dalam Seminar Nasional Teknologi Informasi Kesehatan (SNATIK) 2017, Yogyakarta, 2017.
- [5] A. R. Chrismanto, W. S. Raharjo dan Y. Lukito, "Design and Development of REST-Based Instagram Spam Detector for Indonesian Language," dalam ISEMANTIC 2018, Semarang, 2018.
- [6] R. A. Chrismanto, W. S. Raharjo dan Y. Lukito, "Integration of REST-Based Web Service and Browser Extension for Instagram Spam Detection," International Journal of Advanced Computer Science and Applications, vol. 9, no. 12, December 2018.
- [7] A. Rachmat dan Y. Lukito, "Deteksi Komentar Spam Bahasa Indonesia Pada Instagram Menggunakan Naive Bayes," Ultimatics, vol. 9, no. 1, 2017.
- [8] C. C. Diansheng Guo, "Detecting Non-personal and Spam Users on Geo-tagged Twitter Network," Transaction in GIS, vol. 18, no. 3, pp. 370-384, 2014.
- [9] A. S. S. R. Imam Thoib, "Pengaruh Normalisasi Teks Dengan Text Expansion Dalam Deteksi," Rekayasa Sistem dan Teknologi Informasi, vol. 2, no. 3, pp. 708-715, 2018.
- [10] S. Aiyara dan N. P. Shetty, "N-Gram Assisted Youtube Spam Comment Detection," Procedia Computer Science, vol. 132, pp. 174-182, 8 June 2018.
- [11] D. I. C. P. De Wang, "A social-spam detection framework," dalam Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, Perth, 2011.
- [12] O. W. Ali Akbar Septiandri, "Detecting spam comments on Indonesia's Instagram posts," Journal of Physics: Conference Series (JPCS), vol. 801, no. 1, 2017.
- [13] W. Zhang dan H.-M. Sun, "Instagram Spam Detection," dalam 22nd Pacific Rim International Symposium on Dependable Computing (PRDC), Christchurch, New Zealand, 2018.
- [14] A. Barth, "The Web Origin Concept," December 2011. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc6454.txt>. [Diakses 19 May 2019].
- [15] W. S. Raharjo dan A. Ashari, "IMPLEMENTASI ANNOTEA CLIENT BERBASIS WEB UNTUK MENGATASI ATURAN SAME ORIGIN POLICY," dalam KNASTIK, Yogyakarta, Indonesia, 2009.
- [16] W3Counter, "Browser and Platform Market Share Desember 2018," 31 December 2018. [Online]. Available: <http://www.w3counter.com/globalstats.php?year=2018&month=12>.
- [17] D. Rousset, "Creating One Browser Extension For All Browsers: Edge, Chrome, Firefox, Opera, Brave And Vivaldi," 5 April 2017. [Online]. Available: <https://www.smashingmagazine.com/2017/04/browser-extension-edge-chrome-firefox-opera-brave-vivaldi/>.
- [18] S. A. C. Bukhari, M. Martínez-Romero, M. J. O. Connor, A. L. Egyedi, D. Willrett, J. Graybeal, M. A. Musen, K.-H. Cheung dan S. H. Kleinstejn, "CEDAR OnDemand: a browser extension to generate ontology-based scientific metadata," BMC Bioinformatics, vol. 19, no. 268, 16 July 2018.
- [19] O. Starov dan N. Nikiforakis, "PrivacyMeter: Designing and Developing a Privacy-Preserving Browser Extension,"

- International Symposium on Engineering Secure Software and Systems (ESSoS 2018), pp. 77-95, 2018.
- [20] L. O. deAvelar, G. C. Rezende dan A. P. Freire, "WebHelpDyslexia: A Browser Extension to Adapt Web Content for People with Dyslexia," *Procedia Computer Science*, vol. Volume 67, no. 2015, pp. 150-159, 2015.
- [21] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. %1 dari %2SMC-6, no. 4, pp. 325 - 327, April 1976.
- [22] J. Gou, T. Xiong dan Y. Kuang, "A Novel Weighted Voting for K-Nearest Neighbor Rule," *JOURNAL OF COMPUTERS*, vol. 6, no. 5, pp. 833-840, May 2011.
- [23] J. Gou, L. Du, Y. Zhang dan T. Xiong, "A New Distance-weighted k-nearest Neighbor Classifier," *Journal of Information & Computational Science*, vol. 9, no. 6, p. 1429-1436, June 2012.
- [24] S. B. AKBEN, "A NEW METHOD FOR SELECTION OF NEIGHBORHOOD," *IU-JEE*, vol. 16, no. 1, pp. 2021-2017, 2016.
- [25] Suyanto, *Data Mining untuk Klasifikasi dan Klusterisasi Data*, Bandung: Informatika, 2017.
- [26] M. Aci, C. İnan dan M. Avci, "A hybrid classification method of k nearest neighbor, Bayesian methods and genetic algorithm," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5061-5067, 2010.
- [27] S. Jiang, G. Pang, M. Wu dan L. Kuang, "An improved K-nearest-neighbor algorithm for text categorization," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1503-1509, 2012.
- [28] MDN Web Docs Mozilla, "What are Extension?," 23 October 2018. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/What_are_WebExtensions.
- [29] A. Barth, J. Weinberger dan D. Song, "Cross-Origin JavaScript Capability Leaks: Detection, Exploitation, and Defense," dalam *Usenix Security Symposium*, Montreal, 2009.
- [30] R. Wang, L. Xing, X. Wang dan S. Chen, "Unauthorized origin crossing on mobile platforms: threats and mitigation," dalam *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, Berlin, 2013.