ISSN : 2527-5836
e-ISSN : 2528-0074

Vol. 7 No. 1, Januari 2022

JISKa

Jurnal Informatika Sunan Kalijaga

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta

Tim Pengelola JISKa Edisi Januari 2022

Mitra Bestari (*Reviewer*)

Reviewer Internal:

1. Mandahadi Kusuma, M.Eng. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
2. Maria Ulfa Siregar, Ph.D. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
3. Rahmat Hidayat, M.Cs. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
4. Usfita Kiftiyani, M.Sc. (UIN Sunan Kalijaga Yogyakarta, Indonesia)

Reviewer Eksternal (Mitra Bestari):

1. Ahmad Fathan Hidayatullah, M.Cs. (Universitas Islam Indonesia Yogyakarta, Indonesia)
2. Alam Rahmatulloh, M.T. (Universitas Siliwangi Tasikmalaya, Indonesia)
3. Dr. Cahyo Crysdian (UIN Maulana Malik Ibrahim Malang, Indonesia)
4. Dr. Eng. Ganjar Alfian (Dongguk University Seoul, Korea, Republic of)
5. Muhammad Rifqi Maarif, M.Eng. (Universitas Jenderal Achmad Yani Yogyakarta, Indonesia)
6. Mushab Al Barra, M.Kom. (Universitas Ahmad Dahlan Yogyakarta, Indonesia)
7. Dr.Eng. M. Alex Syaekhoni (Dongguk University Seoul, Korea, Republic of)
8. Norma Latif Fitriyani, M.Sc. (Dongguk University Seoul, Korea, Republic of)
9. Oman Somantri, M.Kom. (Politeknik Negeri Cilacap, Indonesia)
10. Puji Winar Cahyo, M.Cs. (Universitas Jenderal Achmad Yani Yogyakarta, Indonesia)
11. Rischan Mafrur, M.Eng. (The University of Queensland Brisbane, Australia)
12. Suhirman, Ph.D. (Universitas Teknologi Yogyakarta, Indonesia)
13. Yunita Ardilla, M.Sc (Institut Teknologi Sepuluh November Surabaya, Indonesia)

ISSN : 2527-5836

e-ISSN: 2528-0074

JISKa

Vol. 7, No. 1, JANUARI 2022

DAFTAR ISI

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 1, JANUARI, 2022, Pp. 1 – 9
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Analisis Topik Tagar Covidindonesia pada Instagram Menggunakan *Latent Dirichlet Allocation*

Kevin Rafi Adjie Putra Santoso [(1)], Asmaul Husna [(2)], Nadia Widyawati Putri [(3)], Nur Aini Rakhmawati [(4)*]
Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya
e-mail : {kevin.18052,asmaulhusna.18052,nadiaputri.180521}@mhs.its.ac.id, nur.aini@is.its.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 11 April 2021, direvisi 12 Juli 2021, diterima 13 Juli 2021, dan dipublikasikan 25 Januari 2022.

Abstract

In this era, technology is increasingly sophisticated, this is evidenced by the number of people using the internet via cell phones, laptops, and other communication tools. One of the developments of this technology is social media such as Instagram. Along with technological developments, Instagram users can upload and share photos and videos using hashtags (#) so that other users can find the results of their posts. Instagram has now become one of the social media used by more than 1 billion people in the world. In this study, the authors wanted to know the dominant topics discussed through the hashtag covidindonesia. This research was conducted using the Latent Dirichlet Allocation (LDA) method. The analysis was carried out after doing text mining on 84 captions from various users on Instagram. To determine the optimal number of topics, by looking at the value of perplexity and topic coherence. The results obtained are the top 5 topics that are the content material in the uploaded video. These topics include covidindonesia, covid_19, pandemics in Indonesia, and discussion of covid-19 virus mutations.

Keywords: Data Crawling, Instagram, Latent Dirichlet Allocation, Covid Indonesia, Topic Modeling

Abstrak

Pada era sekarang, teknologi semakin canggih, hal ini dibuktikan dengan banyaknya orang yang menggunakan internet melalui telepon genggam, laptop, dan alat komunikasi lainnya, Salah satu perkembangan dari teknologi ini ialah media sosial seperti Instagram. Seiring dengan perkembangan teknologi, pengguna Instagram dapat mengunggah dan membagikan foto dan juga video dengan menggunakan *hashtag* (#) agar pengguna lain dapat menemukan hasil unggahan mereka. Instagram pun kini menjadi salah satu media sosial yang digunakan lebih dari 1 miliar orang di dunia. Pada penelitian ini penulis ingin mengetahui topik dominan yang dibahas melalui *hashtag* covidindonesia. Penelitian ini dilakukan dengan menggunakan metode *Latent Dirichlet Allocation* (LDA). Analisis dilakukan setelah melakukan *text mining* pada 84 *caption* dari berbagai user yang ada di Instagram. Untuk menentukan jumlah topik yang optimal, dengan melihat *nilai perplexity* dan *topic coherence*. Hasil yang didapatkan adalah 5 topik teratas yang menjadi bahan konten dalam video yang diunggah. Topik tersebut antara lain covidindonesia, covid_19, *pandemic* di Indonesia, dan pembahasan mutasi virus covid-19.

Kata Kunci: *Crawling Data*, Instagram, *Latent Dirichlet Allocation*, Covid Indonesia, Pemodelan Topik

1. PENDAHULUAN

Pada era sekarang, teknologi semakin canggih, hal ini dibuktikan dengan banyaknya orang yang menggunakan internet melalui telepon genggam, laptop, dan alat komunikasi lainnya, dengan masifnya perkembangan teknologi, masyarakat dapat dengan mudah memenuhi apa yang dibutuhkannya dalam waktu yang relatif cepat serta mudah dalam hal penggunaannya. Dengan keberadaan internet, secara tidak langsung menghasilkan sebuah generasi yang baru, yaitu generasi *ne(x)t*. Generasi ini dipandang menjadi sebuah generasi masa depan yang diasuh dan

dibesarkan dalam lingkungan budaya baru media digital yang interaktif, yang berwatak menyendiri, kemudian berkomunikasi secara personal, melek komputer atau melek teknologi, dan dibesarkan dengan *videogram* (Ibrahim, 2018). Salah satu perkembangan dari teknologi ialah media sosial seperti Instagram yang muncul pada 6 Oktober 2010. Instagram merupakan salah satu jenis media sosial yang kehadirannya cukup fenomenal, dalam waktu sembilan bulan saja, unggah foto dalam Instagram mencapai angka 150 juta foto di San Francisco. Mengalahkan situs *media-sharing* sejenis Flickr dan situs jejaring Facebook yang fenomenal (Nugraha & Akbar, 2019). Instagram sebagai sebuah media sosial dibangun berdasarkan teknologi Web 2.0 yang membuat penggunanya dapat menyediakan dan berbagi konten (Abdul Talib & Mat Saat, 2017). Pengguna Instagram dapat membagikan foto atau video yang mereka unggah kepada teman dan pengikut mereka. Selain itu, pengguna juga dapat saling berinteraksi dengan melihat, menyukai, dan mengomentari unggahan yang dibagikan di Instagram. Seiring dengan perkembangan teknologi, pengguna Instagram dapat mengunggah dan membagikan foto dan juga video dengan menggunakan *hashtag* (#) agar pengguna lain dapat menemukan hasil unggahan mereka. Instagram pun kini menjadi salah satu media sosial yang digunakan lebih dari 1 miliar orang di dunia (Carman, 2018).

Salah satu topik yang marak dibahas sekarang ialah pandemi covid-19 di Indonesia. Pada awalnya virus ini diduga akibat paparan pasar grosir makanan laut yang banyak menjual banyak spesies hewan hidup. Penyakit ini dengan cepat menyebar di dalam negeri ke bagian lain China. Munculnya 2019-nCoV telah menarik perhatian global tak terkecuali di Indonesia. Pasien pertama yang terkonfirmasi covid-19 di Indonesia berawal dari suatu acara di Jakarta di mana penderita kontak dengan seseorang warga Negara asing (WNA) asal Jepang yang tinggal di Malaysia. Setelah pertemuan tersebut penderita mengeluh demam, batuk dan sesak nafas (Azizah, 2020). Karena penyebaran virus covid-19 yang tidak henti-henti ini akan berdampak secara sosial dan ekonomi. Dalam hal ini Indonesia harus bersiap siaga dalam menghadapinya terutama dalam hal sistem kesehatan yang ada. Melihat pandemi covid-19 yang tak kunjung usai, masyarakat pun banyak memberikan tanggapan mereka melalui Instagram dengan menggunakan *hashtag* covidindonesia.

Berdasarkan uraian di atas, penelitian ini menawarkan solusi untuk melakukan analisis *topic modelling* terhadap topik covid-19 yang dibicarakan di Instagram. *Topic modelling* merupakan metode *non-hierarchical clustering* yang secara otomatis mengklusterkan ke dalam topik yang muncul dari pemodelan sehingga didapatkan topik *cluster* yang sesuai (Alfanzar et al., 2020). Penelitian ini akan mengambil data dari Instagram dengan menggunakan *hashtag* covidindonesia.

Penelitian mengenai Covid-19 sudah pernah dilakukan oleh (Boer et al., 2020) dalam penelitiannya yang berjudul "Analisis Pendapat Siswa Tentang Pembelajaran Berbasis Media Televisi Selama Pandemi Covid-19". Penelitian tersebut bertujuan untuk mengidentifikasi pemberitaan mengenai penanganan virus Covid-19 yang ada di media *online*. Metode yang digunakan adalah metode *framing* di mana para peneliti akan menganalisis suatu fenomena dengan cara memilah kembali dan menyederhanakan berbagai aspek yang menarik perhatian sebagian besar orang. Penelitian ini merupakan penelitian deskriptif kualitatif dengan batasan hanya tiga situs media saja yang dianalisis, yaitu CNNIndonesia.com, Liputan6.com, dan Kompas.com. Sedangkan batasan pada penelitian yang kami lakukan adalah karena kami terbatas pada *caption* Instagram, sehingga terdapat perbedaan karakteristik data yang yang bersumber dari berbagai media sosial, baik dari panjang karakter per dokumen dan sifat kebahasaan yang diterapkan, mengingat *caption* Instagram biasanya menggunakan bahasa informal.

Metode LDA ini juga pernah digunakan oleh (Xue et al., 2020) dalam penelitiannya yang berjudul "*Public discourse and sentiment during the COVID 19 pandemic: Using latent dirichlet allocation for topic modeling on Twitter*". Di mana penelitian dilakukan dengan tujuan untuk memahami wacana dan reaksi psikologis dari pengguna Twitter terhadap Covid-19. Jumlah *tweet* yang dianalisis adalah sekitar 1,9 juta *tweets* berbahasa Inggris yang berkaitan dengan virus corona.

Tweet yang dikumpulkan dimulai dari tanggal 23 Januari hingga 7 Maret 2020. Dalam mengolah dan menganalisis datanya, ada tiga tahapan yang para peneliti lakukan yaitu (1) pengambilan sampel, (2) pengumpulan data, dan (3) *pre-processing* data. Ada beberapa batasan yang terdapat pada penelitian tersebut, yang pertama adalah peneliti hanya mengambil 19 sampel tren *hashtag* dalam pencarian *tweet* untuk penelitian ini, yang kedua bahwa pengguna Twitter tidak merepresentasikan seluruh populasi manusia dan hanya merepresentasikan opini dari orang-orang yang aktif bermain media sosial, dan yang terakhir *tweet* yang menggunakan bahasa selain Bahasa Inggris akan secara otomatis tidak diikutsertakan ke dalam data penelitian.

Metode yang digunakan pada penelitian (Rahmawati et al., 2021) yang berjudul "Analisis topik konten *channel* YouTube K-pop Indonesia menggunakan Latent Dirichlet Allocation". Penelitian dilakukan dengan cara melakukan *crawling data* untuk mengumpulkan data kemudian dilanjutkan dengan Latent Dirichlet Allocation (LDA) dalam klasifikasi datanya. Dalam penelitian itu perhitungan *perplexity* cukup rendah, di mana model LDA yang baik itu dilihat dari nilai *perplexity* yang rendah. *Perplexity* merupakan nilai ukuran statistik tentang seberapa baik model probabilitas memprediksi sampel (Computing for the Social Science, 2021). Batasan yang ada pada penelitian ini adalah sampel hanya diambil dari 10 *channel* Youtube milik warga Indonesia yang berhubungan dengan K-Pop, di mana sampel ini kurang merepresentasikan keseluruhan konten dan media yang ada di Youtube.

Melihat hasil dari penelitian sebelumnya, penulis mengambil kesimpulan bahwa dengan menggunakan metode LDA, peneliti akan menghasilkan akurasi yang sama baiknya dengan penelitian sebelumnya. Dengan menggunakan salah satu metode *topic modeling* yaitu metode LDA, dapat membantu menentukan topik yang dibicarakan pada *hashtag* covidindonesia. LDA mampu meringkas, mengklusterkan, menghubungkan, dan memproses *data text* yang sangat besar, ditambah pula metode yang paling popular dalam *topic modelling* adalah metode LDA.

Penelitian berbasis metode LDA memang sudah banyak dilakukan pada beberapa *platform* media sosial, khususnya aplikasi Twitter. Penerapan metode LDA ini paling banyak ditemukan dan diimplementasikan pada aplikasi media sosial Twitter. Tetapi implementasinya pada *platform* Instagram masih jarang dilakukan, khususnya di Indonesia. Oleh sebab itu, peneliti memilih media sosial Instagram sebagai *platform* yang akan dianalisis mengenai unggahan pengguna yang menggunakan *hashtag* Covid-19. Peneliti memilih Instagram karena aplikasi ini memiliki keunggulan di mana dalam sekali unggahan, pengguna dapat mengunggah *caption*, lokasi, dan juga media berupa foto atau video. Diharapkan penelitian ini dapat memberikan informasi mengenai topik covid-19 yang dibicarakan di Instagram. Selain itu, praktisi akademik dapat menjadikan karya tulis ini sebagai rujukan dari penelitian yang berkaitan.

1.1 Tinjauan Pustaka

1.1.1 *Crawling* Data

Dalam melakukan analisis terhadap sebuah data, tentu diperlukan pengambilan data dari sumber yang diperlukan. Media sosial seperti Instagram dan Twitter kerap menjadi sumber dari data yang ingin dianalisis. Pengambilan data inilah yang disebut dengan *crawling* (Eka Sembodo et al., 2016). Proses dari *crawling* sendiri memerlukan waktu yang tidak sedikit, karena mesin *crawler* harus bisa menjalankan besarnya volume data yang ada dari target *platform* yang diinginkan (Zuliarso & Mustofa, 2009).

1.1.2 *Text Mining*

Text mining merupakan salah satu metode yang digunakan untuk melakukan pengelompokkan atau pengklasifikasian data-data yang berupa tulisan, dokumen, atau teks. *Text mining* termasuk salah satu metode *data mining* yang fokusnya memang untuk mengolah data, teks, atau dokumen dalam jumlah yang besar. Sebelum melakukan *text mining*, perlu dilakukan pemrosesan awal data, baru kemudian data tersebut diolah dengan menggunakan algoritma yang tepat dan sesuai dengan luaran yang diinginkan (Harjanta, 2015).

1.1.3 *Topic Modelling*

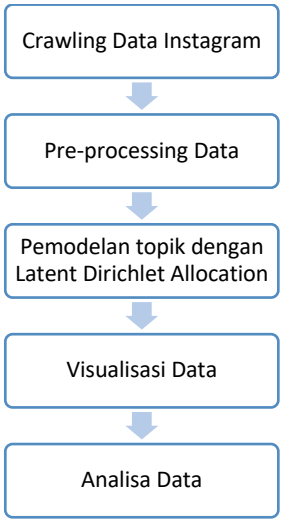
Topic modelling merupakan salah satu metode yang digunakan untuk memodelkan suatu topik dan termasuk salah satu dari *Natural Language Processing* dan memiliki fungsi utama untuk menganalisis suatu teks yang berupa algoritma. Nantinya permodelan ini akan digunakan untuk dapat mengidentifikasi pola-pola yang tidak tampak secara langsung dari serangkaian teks dengan menggunakan metode pendistribusian kata-kata yang ada di dalam teks atau dokumen tersebut. Luaran dari *topic modelling* ini adalah sekumpulan topik yang terdiri dari beberapa kelompok kata berdasarkan pola tertentu yang akan muncul secara bersamaan di dalam teks atau dokumen tersebut. Ada beberapa jenis *topic modelling*, seperti *Latent Semantic Allocation* (LSA), *Probabilistic Latent Semantic Analysis* (PLSA), dan *Latent Dirichlet Allocation* (LDA) (Naury et al., 2021). Jenis yang paling populer saat ini adalah *Latent Dirichlet Allocation* (LDA) karena metode ini cukup mudah dengan melakukan klasterisasi dan meringkas data yang ada, selain itu metode *Latend Dirichlet Allocation* (LDA) ini juga cocok digunakan untuk memproses data yang banyak atau besar (Zulhanif et al., 2017).

1.1.4 *Latent Dirichlet Allocation* (LDA)

Latent Dirichlet Allocation (LDA) merupakan salah satu metode untuk mendeteksi topik dalam sekumpulan data serta melakukan pemodelan topik. *Latent Dirichlet Allocation* (LDA) merupakan metode *topic modeling* dan topik analisis yang paling populer saat ini dan banyak digunakan dalam melakukan analisis pada dokumen yang berukuran sangat besar (Rahmawati et al., 2021) . *Latent Dirichlet Allocation* (LDA) saat ini sering digunakan karena dapat melakukan klasterisasi, melakukan peringkasan, menghubungkan, dan dapat memproses data dengan memberikan bobot pada masing-masing dokumen yang nantinya menghasilkan daftar topik. Ide dasar dari *Latent Dirichlet Allocation* (LDA) ini menganggap bahwa dokumen yang kita ujikan dapat direpresentasikan sebagai sebuah model yang dicampur dari berbagai topik yang dibutuhkan, oleh sebab itu teknik ini disebut sebagai laten (Zulhanif et al., 2017).

2. METODE PENELITIAN

Penelitian ini dimulai dengan mengumpulkan informasi mengenai penelitian serupa yang pernah dilakukan dan studi literatur. Studi literatur didapatkan melalui referensi terpercaya yaitu paper ataupun jurnal, artikel ilmiah, dan *website* resmi yang membahas topik covid indonesia, *topic modelling,* dan *Latent Dirichlet Allocation*. Kemudian dilakukan *crawling data* atau pengumpulan data pada Instagram yakni terkait tagar covidindonesia. Data dikumpulkan dengan menggunakan *command* insta-crawl *hashtag* yang selanjutnya dilakukan pemodelan topik serta analisa. Alur metodologi penelitian disajikan pada Gambar 1.



Gambar 1 Alur Metodologi Penelitian

2.1 *Crawling* Data Instagram

Selanjutnya, dilakukan pengumpulan data atau *crawling* yang digunakan untuk mengumpulkan informasi pada sebuah *website* yaitu Instagram. Pencarian pada Instagram dengan menggunakan *hashtag* covidindonesia akan menjadi dasar pencarian semua *caption* yang kemudian akan dilakukan *indexing* untuk mencari data yang ada pada setiap *post* yang telah diunggah. Data yang dicari ialah data yang terkait dengan *hashtag* covidindonesia, melalui data ini nantinya akan dicari topik apa yang dibahas. Kemudian, akan dilakukan penambangan data berupa *username*, *caption,* dan waktu unggahan *caption* yang akan dihimpun untuk proses penelitian berikutnya.

2.2 *Pre-processing* Data

Data yang telah dihimpun kemudian dilakukan proses *cleaning,* yang mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (Jasmir, 2016). Dari sekumpulan data yang telah didapatkan kemudian dilakukan proses tokenisasi yang bertujuan untuk memisahkan teks ke dalam unit-unit kecil yang biasa disebut dengan token (Rangkuti, 2020). Setelah melakukan tokenisasi, diperlukan pembuangan *stopwords.* Tidak sedikit dalam data teks kita menemukan kata-kata yang sebetulnya tidak diperlukan dalam penelitian. Kata-kata seperti yang, untuk, dari, dan sebagainya tentu bukanlah kata yang berdampak serius pada penelitian bahkan dapat dikatakan tidak dibutuhkan sama sekali.

2.3 Pemodelan Topik dengan *Latent Dirichlet Allocation*

Data yang telah dipisah ke dalam unit-unit kecil kemudian diterapkan ke LDA. Penerapan LDA berfungsi untuk pengenalan pola pada perhitungan statistika dengan cara menemukan proyeksi linear dari data yang akan memaksimalkan jarak antar kelas dan meminimalkan jarak data yang memiliki kesamaan. Tahap pertama dalam LDA ialah mengisi parameter yang akan digunakan sebagai batas data yang digunakan. Kemudian, menerapkan *semi random distribution*, metode ini didasari atas metode distribusi *Dirichlet*. Selanjutnya dilakukan iterasi untuk melihat parameter yang dapat menentukan distribusi dari topik dan kata dari dokumen.

Untuk menentukan banyaknya topik dapat dilakukan dengan menggunakan perhitungan *perplexity*. Nilai *perplexity* didapatkan dari perhitungan dengan menentukan kemungkinan dari log teks dokumen yang tidak terlihat, jika hasil yang didapatkan ialah nilai *perplexity* rendah, dapat dipastikan model tersebut adalah baik. Sementara itu, alur pemodelan topik dengan LDA dimulai dari memuat data yang sudah disimpan ke dalam variabel-variabel tertentu yang sesuai dengan topik penelitian ini yaitu Covid-19, kemudian setelah data-datanya dimuat akan dilanjutkan dengan membentuk model topik dari parameter yang sudah ditentukan, misalnya jumlah kata, jumlah topik, jumlah *hashtag*, dan lain-lain. Kemudian dilakukan pendokumentasian *logging* untuk memantau dan mengawasi aktivitas-aktivitas yang terjadi selama proses pembentukan model dari topik itu sendiri (Putra & Kusumawardani, 2017). Pemodelan topik dengan LDA ini sesuai digunakan untuk menganalisis topik campuran yang memuat beberapa kata sehingga pemodelan topik dengan LDA ini diimplementasikan pada penelitian ini (Nurlayli & Nasichuddin, 2019).

2.4 Visualisasi Data

Data yang didapatkan dari hasil LDA akan menjadi data mentah yang kemudian akan dilakukan visualisasi dengan menggunakan grafik tipe *bar-chart* untuk mempermudah proses analisis data. Dalam visualisasi ini menampilkan perbandingan persebaran topik yang satu dengan yang lainnya.

2.5 Analisa Data

Input dalam proses Analisa data ini ialah data mentah yang didapatkan dari penerapan LDA beserta visualisasi dari hasil LDA yang telah dilakukan pada tahap sebelumnya. Pada proses

analisis ini, data dianalisis menggunakan metode *topic modelling* yang dapat mengelompokkan topik-topik tertentu berdarsarkan kemiripan kata kunci atau pola. Hasil visualisasi data yang sebelumnya berbentuk grafik bertipe *bar-chart* akan dianalisis terkait perbandingan topik yang telah ditemukan. Hasil dari analisis ini akan terus dievaluasi melalui kode program pada zenodo (Husna et al., 2020) untuk menemukan topik-topik yang relevan dan topik yang dibahas dari *hashtag* covidindinonesia. *Source code* yang diimplementasikan untuk menghitung nilai *perplexity* dan nilai *topic coherence* ditampilkan pada Gambar 2.

```
                corpus = [dictionary.doc2bow(text) for text in texts]
                                for k in range(3, 4):
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics=k, id2word = dictionary,
    passes=20, iterations=100, alpha=[0.01]*k, eta=[0.01]*len(dictionary.keys()))
          coherence_model_lda = CoherenceModel(model=ldamodel, texts=texts,
                      dictionary=dictionary, coherence='c_v')
                coherence_lda = coherence_model_lda.get_coherence()
               print(k,ldamodel.log_perplexity(corpus),coherence_lda)
```

Gambar 2 Rumus Perhitungan Nilai *Perplexity* dan Nilai *Topic Coherence*

3. HASIL DAN PEMBAHASAN

3.1 Hasil *Crawling*

Tabel 1 Contoh *Caption Hashtag* Covidindonesia

No.	*Caption* Instagram
1	Selama sepekan terakhir kasus kematian harian di Indonesia terus meningkat,Bahkan se Asia Indonesia peringkat ke 2 setelah India#covid19 #covidindonesia #pandemic #coronaindonesia2w
2	Badan Pemerika Keuangan (BPK) mencatat total anggaran penanganan Covid-19 mencapai Rp 1.035,2 triliun. Auditor Utama Keuangan Negara III BPK Bambang Pamungkas mengatakan, anggaran penanganan Covid-19 itu berasal dari anggaran pendapatan dan belanja negara (APBN) sebesar Rp 937,42 triliun.â€œKemudian dari anggaran pendapatan dan belanja daerah (APBD) sebesar Rp 86,36 triliun dan dari sektor moneter sebesar Rp 6,50 triliun,â€□ katanya dalam konferensi pers via daring, Selasa (29/12).Selain itu, anggaran tersebut juga berasal dari badan usaha milik negara (BUMN) dengan total anggaran sebesar Rp 4,02 triliun. Adapun dari badan usaha milik daerah (BUMD) sekitar Rp 320 miliar dan berasal dari dana hibah dan masyarakat sebesar Rp 625 miliar.@momovele.id#momovele #anggaran #covid19indonesia #covidindonesia #covid_19 #covid19news #ppkm #ppkmdarurat #ppkmjawabali #ppkmdaruratjawabali1h
3	Corona makin menggila.. Semoga ini bermanfaat yahSejumlah cara lain untuk mengaktifkan sistem imun tubuh. Salah satunya menjaga pola makan dengan gizi seimbang.Selain itu, minum air putih sedikitnya 6 gelas/hari, olahraga setidaknya 3 kali dalam seminggu minimal 30 menit, dan menjaga kebersihan tubuh secara keseluruhan dengan mandi setiap hari.Tidak kalah penting, mencuci tangan dengan sabun atau handsanitizer setiap kali akan makan atau minum dan keluar dari kamar mandi, serta istirahat atau tidur yang cukup 6-8 jam/hari.Bagikan tips ini ke keluarga/teman/kerabat kalian.dapatkanMadu Murni dan Rempah Herbaldi @kampusmadu#jsrsehat #imuntubuh #covid #coronavirus #covidindonesia #immunitysistem #jagaimunitas #imunitastubuh #kampusmadu #jsrherbal #resepherbal #herbalcovid #obatcovid #infokesehatan #informasiupdate #updateinfo #infolengkapjsr #zaidulakbar #jsrresep #resepjsr #covidjsr #jintenhitam #ladahitamorganik #maduasli #manfaatmadu #manfatjinten #produkAllah1w

Pada penelitian ini, *crawling* data dilakukan dengan menggunakan *hashtag* covidindonesia pada Instagram. Data yang didapatkan dari pencarian di Instagram dengan *filter* unggahan terpopuler dan terbaru dari beragam pengguna Instagram. Berdasarkan hasil *crawling*, diperoleh 150 *caption* yang memiliki *hashtag* covidindonesia. Tabel 1 merupakan contoh *caption* pada

instagram yang diambil dari tanggal 02/03/2021 sampai 11/07/2021 dengan batasan menggunakan *hashtag* covidindonesia.

3.2 Hasil *Pre-processing* Data

Setelah mengumpulkan data *caption* Instagram dengan proses *crawling*, maka data tersebut tidak dapat langsung digunakan dalam proses analisis. Data yang telah diperoleh harus diproses terlebih dahulu untuk mendapatkan hasil analisis yang maksimal (Husna et al., 2020). Tahap *pre-processing* data ini mencakup membersihkan data dan menghilangkan *stopwords*. Proses tokenisasi juga dilakukan pada data yaitu dengan menggunakan spasi untuk mengganti jarak enter yang ada pada *caption*. Selanjutnya, dilakukan penghapusan *stopwords* atau kata-kata yang tidak memberikan informasi penting untuk penelitian ini. Kata-kata seperti yang, untuk, dari, dan sebagainya dimasukkan menjadi *stopwords* secara manual pada kode dan dihilangkan dari token. Untuk contoh sampel data yang sudah dilakukan proses *preprocessing* dapat dilihat dalam Tabel 2, di mana pada tabel tersebut terdapat 10 sampel data teratas yang telah dilakukan *pre-processing.* Pada Tabel 2 tersebut, penulis memasukkan 3 informasi yaitu *dominant topic,* jumlah *dominant topic*, dan *keywords* yang digunakan pada unggahan di Instagram. Fungsi *len*() juga digunakan dalam kode untuk memisahkan tanda baca yang tidak diperlukan serta selanjutnya dapat dilanjutkan ke proses LDA.

Tabel 2 Sampel Hasil Data *Pre-Processing*

Dominant Topic	**Jumlah *Dominant Topic***	***Keywords***
0,0	7	covidindonesia, covid_19, kerumunan, covid19
1,0	2	covidindonesia, kesehatan, ppkm, covid19
2,0	1	covidindonesia, covid_19, kesehatan, covid19indonesia

3.3 Hasil Penerapan LDA

Pemodelan LDA dilakukan dengan mengubah data menjadi *dictionary* dan *corpus*. Dokumen token yang telah didapatkan melalui tahap *pre-processing*, selanjutnya diubah menjadi sebuah id atau kata kunci dari *dictionary.* Lalu, dokumen token akan diubah menjadi matriks *document-term* untuk selanjutnya dapat membuat model LDA. *Packages* LdaModel digunakan untuk membuat model LDA dan kemudian akan dilakukan perhitungan skor koherensi dari topik menggunakan CoherenceModel. Perhitungan skor koherensi topik juga dibantu oleh *packages* Gensim pada Python.

3.4 Hasil Analisis

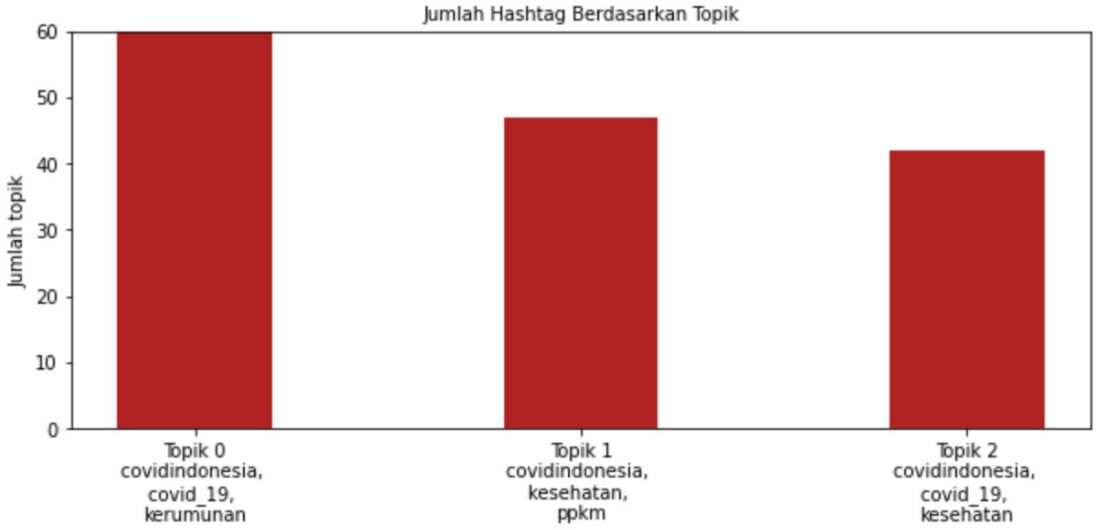
Berdasarkan penerapan LDA yang telah dilakukan pada data, dapat diketahui jumlah topik yang ada pada *caption* Instagram yang mengandung *hashtag* covidindonesia serta nilai *perplexity* dan nilai *topic coherence* yang menunjukkan seberapa baik model yang dihasilkan. Semakin kecil nilai *perplexity* maka semakin bagus model yang dihasilkan. Model yang baik juga memiliki nilai *coherence* yang tinggi. Dengan menggunakan variasi 3, 4, dan 5 topik didapatkan nilai *perplexity* dan nilai *topic coherence* yang berbeda-beda seperti ditunjukkan pada Tabel 3.

Tabel 3 Nilai *Perplexity* dan *Topic Coherence*

Jumlah Topik	**Nilai *Perplexity***	**Nilai *Topic Coherence***
3	-8.704555042489837	0.6555015739681846
4	-8.587164083739	0.6227750210917353
5	-8.501029065460557	0.5541026680947877

Berdasarkan hasil pada Tabel 3, diketahui bahwa nilai *perplexity* dan nilai *coherence* dari model menggunakan 3 topik mengindikasikan bahwa model tersebut lebih baik dibandingkan dengan 4 dan 5 topik. Dengan 3 topik, model memiliki nilai *perplexity* yang sangat kecil dan nilai *topic*

coherence yang besar. Hasil distribusi topik pembicaraan pada *caption* dengan *hashtag* dapat dilihat pada Gambar 3. Grafik pada Gambar 3 menunjukkan persebaran 3 topik yang ada pada *caption* Instagram yang mengandung *hashtag* covidindonesia beserta jumlahnya. Daftar topik yang dominan juga didapatkan dari hasil penerapan LDA seperti disajikan pada tabel Tabel 4.



Gambar 3 Jumlah *Hashtag* Berdasarkan Topik

Tabel 4 Daftar Topik Dominan

No.	*Dominan_Topic*	*Keywords*	Jumlah
1	0,0	covidindonesia, covid_19, kerumunan	66
2	1,0	covidindonesia, kesehatan, ppkm	47
3	2,0	covidindonesia, covid_19, kesehatan	42

Berdasarkan Tabel 4, didapatkan bahwa setiap topik memiliki 3 kata kunci yang berkaitan dengan *hashtag* covidindonesia. Topik 0 memiliki kata kunci *covidindonesia, covid_19*, dan *kerumunan*. Topik 0 memiliki jumlah sebanyak 66. Topik 1 dibentuk dari 47 *caption* yang memiliki kata kunci *covidindonesia, kesehatan*, dan *ppkm*. Pada topik 2 membahas mengenai *covidindonesia, covid_19*, serta *kesehatan*. Topik ini memiliki jumlah yaitu 42.

4. KESIMPULAN

Berdasarkan analisis yang dilakukan dengan menggunakan metode *crawling data* dan *Latent Dirichlet Allocation* (LDA) pada *platform* Instagram, didapatkan bahwa model LDA dengan jumlah topik sebanyak 3 topik merupakan model dengan nilai terbaik. Tiga topik teratas yang berkaitan dengan tagar #covidindonesia di antaranya berupa *hashtag* atau pun suatu kata tertentu. *Hashtag* yang ditemukan di antaranya adalah *covidindonesia, covid_19, kerumunan, ppkm,* dan *kesehatan.* Apabila diurutkan dari topik yang paling dominan, posisi pertama didominasi oleh kata kunci *covidindonesia, covid_19,* dan *kerumunan*. Di posisi kedua didominasi dengan kata kunci c*ovidindonesia, kesehatan,* dan *ppkm.* Sementara untuk posisi ketiga didominasi dengan kata kunci *covidindonesia, covid_19* dan *kesehatan.* Dari hasil analisis dan penelitian yang sudah dilakukan, topik-topik dan kata kunci di atas memang topik dan kata kunci yang relevan dengan tagar *covidindonesia*. Topik yang telah didapatkan melalui metode pemodelan topik *Latent Dirichlet Allocation* (LDA) menunjukkan topik utama yang sedang ramai diperbincangkan. Hal ini dapat bermanfaat bagi pembaca khususnya pengguna media sosial Instagram di Indonesia terkait perkembangan pandemi terkini.

Data yang digunakan dalam penelitian ini bersumber dari *caption* status di Instagram yang memiliki karakteristik bahasa informal sehari-hari. Berdasarkan pengamatan yang dilakukan

penulis, didapatkan dugaan sementara bahwa terdapat perbedaan karakteristik data yang bersumber dari berbagai media sosial, baik dalam hal karakteristik kebahasaan yang diterapkan maupun panjang karakter per dokumen, sehingga hal ini dapat menjadi masukan bagi penelitian selanjutnya.

DAFTAR PUSTAKA

Abdul Talib, Y. Y., & Mat Saat, R. (2017). Social proof in social media shopping: An experimental design research. *SHS Web of Conferences*, *34*, 02005. https://doi.org/10.1051/shsconf/20173402005

Alfanzar, A. I., Khalid, K., & Rozas, I. S. (2020). Topic Modelling Skripsi Menggunakan Metode Latent Diriclhet Allocation. *JSil (Jurnal Sistem Informasi)*, *7*(1), 7. https://doi.org/10.30656/jsii.v7i1.2036

Azizah, K. N. (2020). *Kronologi 2 Pasien Pertama Virus Corona COVID-19 di Indonesia*. DetikHealth. https://health.detik.com/berita-detikhealth/d-4922758/kronologi-2-pasien-pertama-virus-corona-covid-19-di-indonesia

Boer, K. M., Pratiwi, M. R., & Muna, N. (2020). Analisis Framing Pemberitaan Generasi Milenial dan Pemerintah Terkait Covid-19 di Media Online. *Communicatus: Jurnal Ilmu Komunikasi*, *4*(1), 85–104. https://doi.org/10.15575/cjik.v4i1.8277

Carman, A. (2018). *Instagram now has 1 billion users worldwide*. The Verge. https://www.theverge.com/2018/6/20/17484420/instagram-users-one-billion-count

Computing for the Social Science. (2021). *Topic modeling*. Computing for the Social Science. https://cfss.uchicago.edu/notes/topic-modeling/

Eka Sembodo, J., Budi Setiawan, E., & Abdurahman Baizal, Z. (2016). Data Crawling Otomatis pada Twitter. *INDOSC 2016*, *October*, 11–16. https://doi.org/10.21108/INDOSC.2016.111

Harjanta, A. T. J. (2015). Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining. *Jurnal Informatika UPGRIS*, *1*(Juni), 1–9.

Husna, A., Santoso, K. R. A. P., Putri, N. W., & Rakhmawati, N. A. (2020). *asmaul-husna/covidindonesia-LDA: Final | Zenodo*. https://doi.org/10.5281/zenodo.4679443

Ibrahim, I. S. (2018). *Kritik Budaya Komunikasi, Budaya, Media, dan Gaya Hidup Dalam Proses Demokratisasi di Indonesia* (S. O. Pavitasari (ed.)). Jalasutra.

Jasmir. (2016). Implementasi Teknik Data Cleaning dan Teknik Roughset pada Data Tidak Lengkap dalam Data Mining. *Seminar Nasional APTIKOM (SEMNASTIKOM)*, *1*(1), 99–106.

Naury, C., Fudholi, D. H., & Hidayatullah, A. F. (2021). Topic Modelling pada Sentimen Terhadap Headline Berita Online Berbahasa Indonesia Menggunakan LDA dan LSTM. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *5*(1), 24. https://doi.org/10.30865/mib.v5i1.2556

Nugraha, B., & Akbar, M. F. (2019). Perilaku Komunikasi Pengguna Aktif Instagram. *Jurnal Manajemen Komunikasi*, *2*(2), 95. https://doi.org/10.24198/jmk.v2i2.21330

Nurlayli, A., & Nasichuddin, M. A. (2019). Topik Modeling Penelitian Dosen JPTEI UNY pada Google Scholar Menggunakan Latent Dirichlet Allocation. *Elinvo (Electronics, Informatics, and Vocational Education)*, *4*(2), 154–161. https://doi.org/10.21831/elinvo.v4i2.28254

Putra, K. B., & Kusumawardani, R. P. (2017). Analisis Topik Informasi Publik Media Sosial di Surabaya Menggunakan Pemodelan Latent Dirichlet Allocation (LDA). *Jurnal Teknik ITS*, *6*(2). https://doi.org/10.12962/j23373539.v6i2.23205

Rahmawati, A., Nikmah, N. L., Perwira, R. D. A., & Rakhmawati, N. A. (2021). Analisis topik konten channel YouTube K-pop Indonesia menggunakan Latent Dirichlet Allocation. *Teknologi*, *11*(1), 16–25. https://doi.org/10.26594/teknologi.v11i1.2155

Rangkuti, M. D. E. (2020). *Analisis topik komentar video beberapa akun youtube e-commerce indonesia menggunakan metode latent dirichlet allocation*. UIN Syarif Hidayatullah.

Xue, J., Chen, J., Chen, C., Zheng, C., Li, S., & Zhu, T. (2020). Public discourse and sentiment during the COVID 19 pandemic: Using Latent Dirichlet Allocation for topic modeling on Twitter. *PLOS ONE*, *15*(9), e0239441. https://doi.org/10.1371/journal.pone.0239441

Zulhanif, Sudartianto, Tantular, B., & Jaya, I. G. N. M. (2017). Aplikasi Latent Dirichlet Allocation (LDA) pada Clustering Data Teks. *Jurnal Logika*, *7*(1), 46–51.

Zuliarso, E., & Mustofa, K. (2009). Crawling Web berdasarkan Ontology. *Jurnal Teknologi Informasi DINAMIK*, *XIV*(2), 105–112. https://doi.org/10.35315/dinamik.v14i2.97

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 1, JANUARI, 2022, Pp. 10 – 19
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Analisis Sentimen *Tweet* Tentang UU Cipta Kerja Menggunakan Algoritma SVM Berbasis PSO

Trifebi Shina Sabrila [(1)*], Yufis Azhar [(2)], Christian Sri Kusuma Aditya [(3)]
Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Malang, Malang
e-mail : trifebiss@webmail.umm.ac.id, {yufis,christiaskaditya}@umm.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 12 Juni 2021, direvisi 26 September 2021, diterima 21 Oktober 2021, dan dipublikasikan 25 Januari 2022.

Abstract

Support Vector Machine (SVM) is one of the most widely used classification algorithms for sentiment analysis and has been shown to provide satisfactory performance. However, despite its advantages, the SVM algorithm still has weaknesses in selecting the right SVM parameters to optimize the performance. In this study, sentiment analysis was done with the use of data called tweets about Undang-Undang Cipta Kerja which reap many pros and cons by the people in Indonesia, especially the laborers. The classification method used in this study is the Support Vector Machine algorithm which is optimized using the Particle Swarm Optimization method for the SVM parameters selection in the hope of optimizing the performance generated by the SVM algorithm in sentiment analysis. The results of the study using 10 k-fold cross-validations using the SVM algorithm resulted in an accuracy of 92,99%, a precision of 93,24%, and a recall of 93%. Meanwhile, the SVM and PSO algorithms produce an accuracy of 95%, precision of 95,08%, and recall of 94,97%. The results show that the Particle Swarm Optimization method can overcome the weaknesses of the Support Vector Machine algorithm in the problem of parameter selection and has succeeded in improving the resulting performance where the SVM-PSO is more superior to SVM without optimization in sentiment analysis.

Keywords: Sentiment Analysis, Twitter, UU Cipta Kerja, SVM, PSO

Abstrak

Support Vector Machine (SVM) tergolong sebagai satu di antara algoritma klasifikasi yang paling sering dimanfaatkan dalam analisis sentimen dan terbukti memberikan performa yang memuaskan. Akan tetapi, terlepas dari keunggulan yang diberikan, algoritma SVM masih mempunyai kelemahan dalam penentuan parameter SVM yang tepat untuk mengoptimalkan performa yang dihasilkan. Pada penelitian ini, dilakukan analisis sentimen pada *tweet* tentang Undang-Undang Cipta Kerja yang menuai banyak pro dan kontra oleh masyarakat di Indonesia terutama bagi para buruh. Metode klasifikasi yang diterapkan pada penelitian ialah algoritma *Support Vector Machine* yang dioptimasi menggunakan metode *Particle Swarm Optimization* sebagai seleksi pada parameter SVM dengan harapan dapat mengoptimalkan performa yang dihasilkan oleh algoritma *Support Vector Machine* dalam analisis sentimen. Hasil penelitian menggunakan 10 *k-fold cross validation* menggunakan algoritma SVM menghasilkan akurasi sebesar 92,99%, presisi sebesar 93,24%, dan *recall* sebesar 93%. Sementara pada algoritma SVM dan PSO menghasilkan akurasi sebesar 95%, presisi sebesar 95,08%, dan *recall* sebesar 94,97%. Hasil menunjukkan bahwa metode optimasi *Particle Swarm Optimization* dapat mengatasi kelemahan algoritma *Support Vector Machine* dalam masalah pemilihan parameter dan berhasil meningkatkan performa yang dihasilkan di mana SVM-PSO lebih unggul dibandingkan SVM biasa dalam analisis sentimen.

Kata Kunci: Analisis Sentimen, Twitter, UU Cipta Kerja, SVM, PSO

1. PENDAHULUAN

Pertumbuhan media sosial belakangan ini amat berpengaruh serta memberikan peran besar bagi beragam aspek kehidupan masyarakat di era digital modern ini. Twitter merupakan salah satu media sosial terpopuler karena kemudahan dalam penggunaan dan pengaksesannya. Hingga saat ini, diperkirakan Twitter telah mencapai 330 juta pengguna aktif setiap bulannya dan terus

meningkat setiap harinya (Rekha et al., 2019). Melalui Twitter, pengguna dapat melakukan berbagai macam aktivitas, komunikasi antar individu maupun kelompok, menuliskan kegiatan sehari-hari, mempromosikan dagangan, beradu argumen, hingga menyampaikan opini terkait suatu topik hanya dengan membuat unggahan berupa pesan status (biasa disebut *tweet*) yang dibatasi hingga 280 karakter. *Tweet* yang dituliskan oleh pengguna tersebut merupakan sumber yang bisa dimanfaatkan untuk menganalisis sentimen masyarakat umum terhadap isu yang dibahas karena tulisan tersebut berisikan sentimen yang dapat dimanfaatkan sebagai sumber evaluasi dan pertimbangan dalam menarik suatu keputusan ke depannya.

Pada awal Oktober 2020 lalu, pengesahan UU Cipta Kerja oleh DPR RI menjadi salah satu persoalan yang menuai banyak perbincangan di dunia nyata bahkan di media sosial terlebih Twitter. Pengesahan UU Cipta Kerja ini menuai berbagai reaksi pro dan kontra dalam masyarakat terutama para buruh. Masyarakat pun dengan bebas menyampaikan opininya baik berupa penolakkan atau dukungan dalam menanggapi keputusan DPR RI dalam pengesahan UU ini melalui *tweet.* Hal ini menarik perhatian untuk dilakukannya analisis sentimen terhadap *tweet* opini masyarakat mengenai pengesahan UU Cipta Kerja untuk mengetahui kecenderungan respon masyarakat terkait isu tersebut.

Analisis sentimen adalah proses penggalian, pemahaman, dan pengekstraksian informasi dari data berupa teks yang digunakan untuk mengdentifikasi sentimen yang terdapat dalam sebuah kalimat opini secara otomatis (Hayatin et al., 2020; Iqbal et al., 2019) dengan menggunakan metode *Natural Language Processing* (NLP), statistik, atau *Machine Learning* (ML), yang selanjutnya akan diklasifikasikan ke dalam polaritas positif ataupun negatif (Sharma & Daniels, 2020).

Algoritma *Support Vector Machine* (SVM) ialah satu di antara algoritma klasifikasi yang paling banyak dimanfaatkan dalam klasifikasi teks termasuk analisis sentimen dan terbukti memberikan performa yang lebih unggul dibandingkan algoritma klasifikasi lainnya (Basari et al., 2013). Namun, terlepas dari baiknya performasi yang diberikan, algoritma *Support Vector Machine* (SVM) masih mempunyai kelemahan dalam penentuan parameter dan fitur yang tepat untuk mengoptimalkan performa yang dihasilkan (Kristiyanti & Wahyudi, 2017; Windha Mega & Haryoko, 2019). Oleh karena itu, diperlukan metode optimasi untuk melakukan pemilihan parameter SVM yang tepat sehingga dapat menangani kelemahan algoritma tersebut.

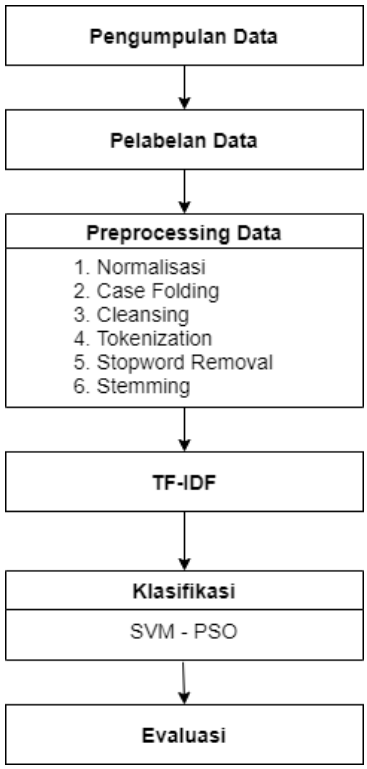
Pada beberapa penelitian sebelumnya, analisis sentimen terhadap data Twitter dilakukan oleh Godara & Kumar (2019) dengan membandingkan lima jenis algoritma klasifikasi, yaitu *K-Nearest Neighbor* (KNN), *Decision Tree*, *Artificial Neural Network* (ANN), *Naive Bayes*, dan *Support Vector Machine* (SVM) terhadap dataset Twitter. Hasil uji coba memperlihatkan algoritma SVM mengungguli metode lainnya dengan hasil akurasi 92,34%. Penelitian lain juga dilakukan oleh Tineges et al. (2020) tentang analisis sentimen pada *tweet* terkait layanan IndiHome dengan memanfaatkan algoritma SVM serta metode ekstraksi fitur TF-IDF dengan hasil akurasi sebesar 87%. Selain itu, Rustam et al. (2019) melakukan klasifikasi sentimen pada tweet dengan melakukan perbandingan terhadap tiga metode ekstraksi fitur, TF, TF-IDF, dan Word2Vec pada algoritma pembelajaran mesin. Hasil pengujian menunjukkan bahwa TF-IDF merupakan metode ekstraksi fitur yang terbaik dalam penelitian ini. Penelitian terkait analisis sentimen terhadap *review* jasa maskapai penerbangan dengan memanfaatkan algoritma SVM berbasis PSO dikerjakan oleh Risawati et al. (2020) mendapatkan akurasi sebesar 87,34% yang mana lebih unggul jika dibandingkan hanya dengan menggunakan algoritma SVM saja dengana kurasi sebesar 84,25%. Berikutnya, Arsi et al. (2021) juga melakukan penelitian dengan menggunakan algoritma SVM yang dioptimasi dengan metode PSO mengenai analisis sentimen wacana pindah Ibu Kota Indonesia dengan nilai akurasi yang diperoleh sebesar 81,15%, yakni lebih unggul 2,09% jika dibandingkan hanya menggunakan algoritma SVM saja yang hanya memperoleh akurasi sebesar 79,06%. Analisis sentimen terhadap opini masyarakat mengenai transportasi *online* di Twitter juga dilakukan oleh Que et al. (2020) dengan menerapkan algoritma SVM dan metode optimasi PSO yang menjadi rujukan utama dalam penelitian ini. PSO terbukti sanggup untuk mengatasi kelemahan SVM terhadap pemilihan parameter yang tepat dalam klasifikasi

sentimen dengan meningkatkan hasil akurasi sebesar 96,04%, di mana akurasi tersebut unggul sebesar 0,58% jika dibandingkan dengan algoritma *Support Vector Machine* saja tanpa optimasi sebesar 95,46%.

Berdasarkan permasalahan yang telah dijabarkan sebelumnya, maka akan dilakukan penelitian terkait analisis sentimen terhadap *tweet* berbahasa Indonesia yang mengandung opini atau komentar masyarakat mengenai pengesahan UU Cipta Kerja oleh DPR RI dengan menerapkan algoritma *Support Vector Machine* (SVM) dan diintegrasikan dengan metode optimasi *Particle Swarm Optimization* (PSO) dengan harapan dapat mengoptimalkan performa yang dihasilkan oleh algoritma *Support Vector Machine* dalam proses analisis sentimen.

2. METODE PENELITIAN

Pada bagian ini, akan diuraikan langkah yang akan dilaksanakan pada penelitian. Secara garis besar alur penelitian ditampilkan pada Gambar 1.



Gambar 1 Alur Penelitian

2.1 Pengumpulan Data

Penelitian ini memanfaatkan data opini yang bersumber dari Twitter berupa *tweet* berbahasa Indonesia dengan topik UU Cipta Kerja. Proses pengumpulan data diproses dengan teknik *crawling* menggunakan 2 macam *tools,* yaitu Twitter API dan *snscrape* dengan total data sebanyak 1000 tweet.

2.2 Pelabelan Data

Data yang telah berhasil dikumpulkan kemudian akan diberikan label pada setiap datanya. Kategori yang digunakan dalam penelitian ini hanya terdiri atas label positif bagi data yang mengandung opini positif dan label negatif untuk data yang mengandung opini negatif. Proses pelabelan data ini dilakukan secara manual dengan bantuan 3 orang anotator yang terdiri atas 2 anotator primer dan 1 anotator sekunder.

2.3 Preprocessing Data

Preprocessing adalah salah satu tahap yang paling penting di dalam *text mining* (Alasadi & Bhaya, 2017) yang digunakan untuk melakukan pengubahan data mentah menjadi data yang siap digunakan dan lebih terstruktur dengan pembersihan dan penyeragaman data karena data yang dikumpulkan biasanya masih berupa data kotor. Adapun tahap *preprocessing* yang diterapkan dalam penelitian ini adalah normalisasi, *case folding, cleansing, tokenization, stopword removal,* dan *stemming.*

2.3.1 Normalisasi

Normalisasi adalah proses yang bertujuan untuk memperbaiki kata yang memiliki kesalahan penulisan ataupun pengejaan serta kata yang dituliskan dengan disingkat.

2.3.2 Case Folding

Case folding adalah teknik guna mengganti huruf kapital yang ditemukan pada data sebagai huruf kecil seluruhnya.

2.3.3 Cleansing

Cleansing adalah proses yang bertujuan untuk menghilangkan berbagai informasi yang tidak dibutuhkan dalam proses analisis sentimen baik berupa *link* (http, https, pic.twitter), *hashtag*, *username* (dituliskan *@username*), maupun karakter spesial lainnya untuk memperoleh hasil analisis yang lebih baik.

2.3.4 Tokenization

Tokenization adalah proses pemecahan setiap kalimat yang terdapat pada data menjadi potongan-potongan kata. Caranya adalah dengan menjadikan spasi sebagai acuan untuk memisah setiap katanya.

2.3.5 Stopword Removal

Stopword ialah kata yang tidak bermakna penting, seperti kata “di”, “dan”, “dengan”, “oleh”, dan sebagainya. Sehingga, *stopword removal* ialah proses guna meniadakan kata yang tidak memiliki makna berharga terhadap data yang akan digunakan.

2.3.6 Stemming

Stemming adalah proses untuk meniadakan imbuhan yang didapati pada suatu kata menjadi sebuah kata dasar (Prasetyaningrum et al., 2020). Proses ini memiliki tujuan untuk mengurangi banyaknya varians dari suatu kata agar tidak menambah beban memori.

2.4 *Term Frequency-Inverse Document Frequency* (TF-IDF)

Setelah dilakukan tahap *preprocessing*, data akan masuk ke dalam proses *term weighting* untuk dilakukan proses pembobotan atau pemberian nilai pada setiap *term* yang terdapat pada data. Salah satu metode *term weighting* yang paling sering diterapkan dalam *text mining* dan akan dimanfaatkan pada penelitian ini adalah *Term Frequency-Inverse Document Frequency* (TF-IDF).

TF-IDF adalah teknik untuk memberikan bobot pada kata guna mengekstraksi setiap kata pada *dataset* yang menggabungkan konsep *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF), di mana TF ialah total kemunculan *term* di suatu dokumen, semakin tinggi total suatu *term* yang ada di suatu dokumen, maka akan semakin dianggap penting pula dokumen tersebut (Jo, 2019), sedangkan IDF memperlihatkan kebersangkutan dari ketersediaan suatu *term* pada keseluruhan dokumen, semakin minim frekuensi dokumen yang memuat *term* yang

diimplikasikan, nilai IDF pun akan semakin tinggi. Rumus TF-IDF yang digunakan dapat dilihat pada persamaan berikut (Jo, 2019):

$$W_{ij} = tf_{ij} \times idf_j = tf_{ij} \times log\frac{N}{df_j} \quad (1)$$

Di mana W_{ij} adalah bobot *term* ke-j terhadap dokumen-i yang akan kita cari dengan tf_{ij} sebagai frekuensi kemunculan *term*-j dalam dokumen-i, N berupa total dokumen secara keseluruhan, dan df_j berupa total dokumen yang mengandung *term*-j.

2.5 Klasifikasi

Setelah seluruh tahapan sebelumnya selesai dilakukan, selanjutnya akan dilakukan proses klasifikasi sentimen terhadap data yang sudah disiapkan. Klasifikasi yang dilakukan pada penelitian dilakukan dengan dua macam proses klasifikasi, yang pertama adalah klasifikasi sentimen dengan metode SVM saja, dan yang kedua adalah klasifikasi sentimen dengan metode SVM yang diintegrasikan dengan metode PSO.

2.5.1 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah salah satu teknik klasifikasi yang tergolong sebagai *supervised learning* dan merupakan bagian dari pembelajaran mesin yang prosesnya mengikuti aturan *Structural Risk Minimization* (SRM) guna mendapatkan *hyperplane* paling optimal dengan margin optimumnya yang membagi setiap kelas yang ada di dalam ruang *input* (Wardhani et al., 2018). SVM juga merupakan metode yang mampu mengatasi permasalahan secara *linier* maupun *non-linier*. Pada dasarnya, ketika memprediksi suatu kelas pada data, SVM akan memberikan label berdasarkan daerah kelas mana yang ditempati oleh data tersebut (Rana & Singh, 2016). Letak awal *hyperplane* pada SVM ditentukan oleh nilai dari parameter SVM yang diinisialisasikan di awal seperti nilai C, gamma, jenis kernel, dan sebagainya, sehingga pemilihan parameter SVM yang tepat sangatlah berpengaruh dengan performa yang akan dihasilkan.

2.5.2 *Particle Swarm Optimization* (PSO)

Particle Swarm Optimization (PSO) ialah suatu metode optimasi yang berdasar pada populasi atas sekumpulan partikel dengan kecepatan dan posisi yang selalu diperbarui dalam setiap iterasinya (Kiranyaz et al., 2014). PSO biasanya banyak digunakan dalam pemecahan masalah optimasi dan masalah seleksi fitur (Liu et al., 2011). Setiap partikel pada PSO akan melacak posisi di dalam ruang pencarian serta solusi terbaiknya yang disebut dengan *personal best (pbest)* serta *global best (gbest)* yang dicapai oleh populasi dengan indeks partikelnya (Kiranyaz et al., 2014). Persamaan yang digunakan untuk *update* kecepatan dan posisi dalam PSO ialah seperti berikut:

$$V_i(t) = \theta V_{i(t-1)} + c_1 r_1 \left[Pbest_i - X_{i(t-1)}\right] + c_2 r_2 \left[Gbest - X_{i(t-1)}\right] \quad (2)$$

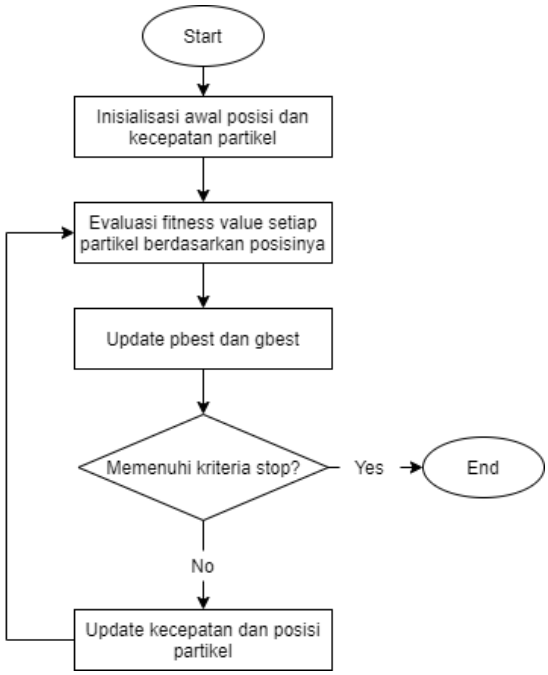
Di mana $V_i(t)$ adalah kecepatan ke t, θ adalah nilai inersia, $c_1 c_2$ adalah *learning rate*, $r_1 r_2$ adalah bilangan *random* dengan *range* 0 – 1, dan $X_{i(t-1)}$ adalah posisi partikel sebelumnya. Setelah mendapatkan $V_i(t)$ sebagai kecepatan yang baru, selanjutnya posisi partikel yang baru akan dihitung dengan menggunakan persamaan berikut:

$$X_i(t) = V_i(t) + X_i(t-1) \quad (3)$$

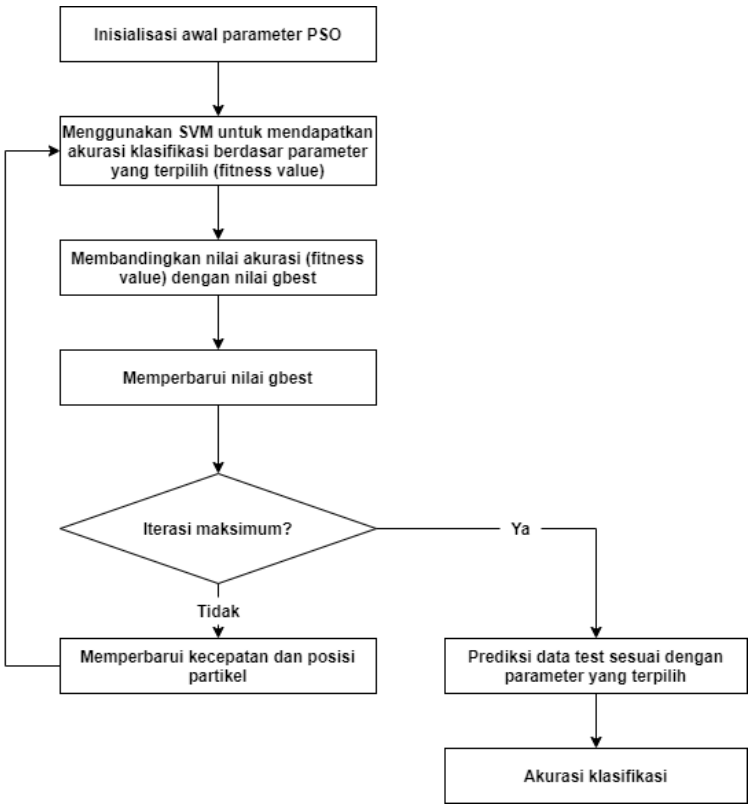
Di mana $X_i(t)$ adalah posisi partikel yang baru, $V_i(t)$ adalah kecepatan yang baru, dan $X_i(t-1)$ adalah posisi partikel yang sebelumnya. Alur kerja PSO sendiri secara umum dapat dilihat pada diagram alur di Gambar 2.

Sementara itu, untuk penerapan metode PSO sebagai seleksi parameter SVM pada klasifikasi menggunakan algoritma SVM pada penelitian ini, algoritma SVM akan mendapatkan hasil akurasi

klasifikasi berdasarkan nilai parameter SVM yang telah terpilih menggunakan metode PSO. Parameter SVM yang akan dioptimasi pada penelitian ini adalah nilai C dan nilai Gamma. Gambar 3 menunjukkan langkah-langkah untuk mengoptimalkan SVM menggunakan PSO dalam analisis sentimen (Hayatin et al., 2020).



Gambar 2 Alur Kerja *Particle Swarm Optimization*



Gambar 3 Alur Model Klasifikasi SVM-PSO

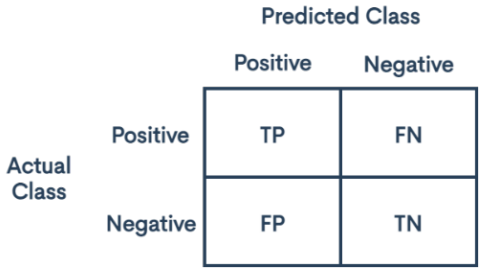
Pada tahap pertama, dilakukan inisialisasi awal parameter PSO. Selanjutnya akan dilakukan evaluasi *fitness value* menggunakan algoritma SVM untuk mendapatkan akurasi klasifikasi berdasarkan parameter tiap partikel yang telah terpilih. Jika proses belum mecapai iterasi maksimumnya, maka kecepatan dan posisi tiap partikel akan terus diperbarui dengan menggunakan Persamaan (2) dan (3) hingga selesai diproses dan mencapai iterasi maksimum serta mencapai nilai akurasi yang terbaik untuk kemudian digunakan sebagai model untuk menguji data *test* sesuai dengan parameter yang terpilih.

2.6 Evaluasi

Hasil dari klasifikasi yang berhasil dilakukan berikutnya masuk ke dalam tahap evaluasi dengan tujuan untuk menguji dan mengetahui performa dari model yang telah diusulkan. Evaluasi yang dilaksanakan pada penelitian ini adalah melalui perhitungan nilai akurasi, presisi, serta *recall* sesuai dengan nilai yang terdapat pada *confusion matrix* berdasarkan pengujian model yang diproses dengan menerapkan *k-fold cross validation* dengan nilai k sebesar 10.

K-fold cross validation adalah metode guna melaksanakan evaluasi performa suatu model di mana data dipecah ke dalam dua bagian yaitu data *train* dan data *test*. *K-fold cross validation* dengan nilai k sebesar 10, data akan dipecah ke dalam 10 bagian data di mana 9 bagian akan berperan sebagai data latih dan satu bagian akan berperan sebagai data uji, proses ini dilakukan secara bergantian sebanyak 10 kali menggunakan model yang telah dibentuk.

Sedangkan *confusion matrix* sendiri adalah konsep dalam pembelajaran mesin yang mengandung informasi mengenai nilai yang sesungguhnya serta prediksi hasil klasifikasi yang sudah diproses oleh suatu sistem (Deng et al., 2016). *Confusion matrix* digunakan untuk mengevaluasi kinerja dari sistem yang telah dibangun. Contoh dari *confusion matrix* ditampilkan pada Gambar 4.



Gambar 4 Contoh *Confusion Matrix*

Terdapat istilah nilai hasil klasifikasi yang terdapat dalam *confusion matrix*, yaitu *True Positive* (TP) yang memberikan banyak data positif yang berhasil diprediksi secara benar, *True Negative* (TN) yang memberikan banyak data negatif yang diprediksi secara benar, *False Positive* (FP) yang memberikan banyak data positif namun diprediksikan salah, dan *False Negative* (FN) yang memberikan banyak data negatif namun diprediksikan salah (Al Rivan et al., 2020). Berdasarkan data dari *confusion matrix* tersebut, kinerja suatu sistem dapat diketahui dengan melakukan perhitungan terhadap nilai akurasi, presisi, dan *recall* dari sistem tersebut dengan menggunakan persamaan berikut (Al Rivan et al., 2020):

$$Akurasi = \frac{TP + TN}{TP + TN + FP + F\,N} \times 100\%. \quad (4)$$

$$Presisi = \frac{TP\,.}{TP + FP} \times 100\,\% \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \times 100\,\% \quad (6)$$

3. HASIL DAN PEMBAHASAN

Pada bab ini akan dipaparkan hasil dari penelitian yang telah dilaksanakan. Tahap pengujian pada penelitian menerapkan bahasa pemrograman *python* untuk proses implementasinya.

3.1 Dataset

Dataset yang dimanfaatkan pada penelitian ini adalah data berupa *tweet* berbahasa Indonesia yang memuat topik mengenai UU Cipta Kerja. Pengumpulan data dikerjakan dengan proses *crawling* menggunakan dua macam *tools,* yaitu Twitter API dan *snscrape*. Dataset yang dikumpulkan berjumlah 1000 data. Proses pelabelan data dikerjakan secara manual oleh tiga orang anotator yang merupakan ahli bahasa Indonesia sesuai dengan bahasa pada data yang dikumpulkan, yang mana bertujuan agar interpretasi maksud atas setiap teks dapat masuk ke dalam kelas sentimen yang sesuai. Setelah dilaksanakan pelabelan data, sebanyak 500 data masuk ke dalam label positif dan 500 data masuk ke dalam label negatif.

3.2 Hasil Klasifikasi

Pengujian klasifikasi dalam penelitian ini dilakukan dalam empat skenario pengujian dengan menggunakan 10 *fold cross validation*. Di mana dalam skenario pengujian kesatu, data yang telah siap digunakan diproses untuk klasifikasi sentimen menggunakan algoritma *Logistic Regression* yang mana ialah satu di antara metode klasifikasi yang sering dipergunakan dalam klasifikasi teks sebagai pembanding di antara algoritma SVM dan SVM-PSO. Pada skenario pengujian kedua, dilakukan proses pengujian dengan menggunakan algoritma *Support Vector Machine* tanpa optimasi dengan menggunakan parameter *default* SVM. Pada skenario pengujian ketiga, dilakukan proses pengujian dengan menggunakan algoritma SVM dan metode optimasi PSO sebagai pemilihan parameter SVM dengan *setting* parameter untuk SVM dan PSO secara *default* seperti yang dilakukan pada penelitian sebelumnya. Kemudian pada skenario pengujian keempat, dilakukan proses pengujian dengan menggunakan algoritma SVM dan metode optimasi PSO sebagai pemilihan parameter SVM dengan iterasi sebanyak 50 kali.

Berdasarkan pengujian yang telah dilakukan pada keempat skenario, didapatkan perbandingan hasil perhitungan akurasi, rata-rata presisi, dan rata-rata *recall* yang ditampilkan pada Tabel 1.

Tabel 1 Perbandingan Hasil Evaluasi

Skenario	Akurasi	Presisi	*Recall*
Logistic Regression	91,70%	91,90%	91,70%
SVM tanpa optimasi	92,99%	93,24%	93%
SVM-PSO parameter *default* (Que et al., 2020)	94,89%	95%	94,88%
SVM-PSO 50 iterasi	95%	95,08%	94,97%

Pada tabel hasil evaluasi klasifikasi sentimen empat skenario di atas diperoleh nilai akurasi, pada skenario pengujian pertama dengan menggunakan algoritma *Logistic Regression* sebesar 91,70%, presisi sebesar 91,90%, dan *recall* sebesar 91,70. Selanjutnya pada skenario pengujian kedua diperoleh nilai akurasi sebesar 92,99%, presisi sebesar 93,24%, dan *recall* sebesar 93% di mana pada skenario ini proses klasifikasi hanya dilakukan menggunakan algoritma SVM tanpa optimasi menggunakan metode PSO. Kemudian, diperoleh nilai akurasi pada skenario pengujian ketiga yakni menggunakan SVM-PSO dengan parameter *default* sebesar 94,89%, presisi sebesar 95%, dan *recall* sebesar 94,88%. Terakhir pada skenario keempat yakni menggunakan SVM-PSO dengan iterasi PSO 50 kali didapatkan akurasi sebesar 95%, presisi sebesar 95,08%, dan *recall* sebesar 94,97%.

Berdasarkan hasil yang diperoleh tersebut, dapat terlihat bahwa pengujian dengan menggunakan algoritma *Support Vector Machine* baik yang hanya menggunakan *Support Vector Machine* saja ataupun yang dioptimasi menggunakan metode *Particle Swarm Optimization* mendapatkan hasil yang lebih tinggi jika dibandingkan dengan pengujian menggunakan algoritma *Logistic*

Regression. Hal berikut membuktikan bahwa algoritma SVM memang lebih unggul ketika digunakan dalam klasifikasi teks. Selanjutnya, algoritma SVM yang telah dioptimasi dengan PSO dapat meningkatkan performa yang dihasilkan dalam seluruh parameter pengujian jika dibandingkan dengan algoritma SVM saja tanpa optimasi dengan kenaikan akurasi sebesar 1,9%, presisi sebesar 1,76%, dan *recall* sebesar 1,88% ketika parameter PSO di *setting default,* dan kenaikan akurasi sebesar 2,01%, presisi sebesar 1,84%, dan *recal*l sebesar 1,97% ketika iterasi PSO di *setting* sebanyak 50 kali. Hal tersebut menunjukkan bahwa PSO benar dapat meningkatkan perfoma yang dihasilkan oleh algoritma SVM dengan melakukan optimasi serta mengatasi kelemahan dalam pemilihan parameter SVM-nya sesuai dengan yang dinyatakan pada sejumlah penelitian terdahulu.

4. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan terkait analisis sentimen pada *tweet* tentang UU Cipta Kerja, didapatkan hasil evaluasi pada skenario pengujian pertama klasifikasi menggunakan algoritma SVM saja tanpa optimasi dengan nilai akurasi 92,99%, nilai rata-rata presisi 93,24%, dan nilai rata-rata *recall* 93%. Sedangkan untuk klasifikasi menggunakan algoritma SVM yang diintegrasikan dengan PSO untuk pemilihan parameter SVM, hasil evaluasi pada skenario pengujian kedua mendapat hasil akurasi sebesar 95%, nilai rata-rata presisi sebesar 95,08%, dan nila rata-rata *recall* sebesar 94,97%.

Dari proses klasifikasi tersebut dapat disimpulkan bahwa metode optimasi *Particle Swarm Optimization* dapat mengatasi kelemahan algoritma *Support Vector Machine* dalam masalah pemilihan parameter dan meningkatkan seluruh parameter pengujian baik nilai akurasi, presisi, maupun *recall* terhadap model yang dibangun.

DAFTAR PUSTAKA

Al Rivan, M. E., Rachmat, N., & Ayustin, M. R. (2020). Klasifikasi Jenis Kacang-Kacangan Berdasarkan Tekstur Menggunakan Jaringan Syaraf Tiruan. *Jurnal Komputer Terapan*, *6*(1), 89–98. https://doi.org/10.35143/jkt.v6i1.3546

Alasadi, S. A., & Bhaya, W. S. (2017). Review of Data Preprocessing Techniques in Data Mining. In *Journal of Engineering and Applied Sciences* (Vol. 12, Issue 16, pp. 4102–4107). https://doi.org/10.3923/jeasci.2017.4102.4107

Arsi, P., Wahyudi, R., & Waluyo, R. (2021). Optimasi SVM Berbasis PSO pada Analisis Sentimen Wacana Pindah Ibu Kota Indonesia. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(2), 231–237. https://doi.org/10.29207/resti.v5i2.2698

Basari, A. S. H., Hussin, B., Ananta, I. G. P., & Zeniarja, J. (2013). Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization. *Procedia Engineering*, *53*, 453–462. https://doi.org/10.1016/j.proeng.2013.02.059

Deng, X., Liu, Q., Deng, Y., & Mahadevan, S. (2016). An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, *340–341*, 250–261. https://doi.org/10.1016/j.ins.2016.01.033

Godara, N., & Kumar, S. (2019). Opinion Mining using Machine Learning Techniques. *International Journal of Engineering and Advanced Technology*, *9*(2), 4287–4292. https://doi.org/10.35940/ijeat.B4108.129219

Hayatin, N., Marthasari, G. I., & Nuarini, L. (2020). Optimization of Sentiment Analysis for Indonesian Presidential Election using Naive Bayes and Particle Swarm Optimization. *Jurnal Online Informatika*, *5*(1), 81–88. https://doi.org/10.15575/join.v5i1.558

Iqbal, F., Hashmi, J. M., Fung, B. C. M., Batool, R., Khattak, A. M., Aleem, S., & Hung, P. C. K. (2019). A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction. *IEEE Access*, *7*(c), 14637–14652. https://doi.org/10.1109/ACCESS.2019.2892852

Jo, V. (2019). Introduction. *Seminars in Diagnostic Pathology*, *36*(2), 83–84. https://doi.org/10.1053/j.semdp.2019.02.002

Kiranyaz, S., Ince, T., & Gabbouj, M. (2014). *Multidimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition*. Springer.

Kristiyanti, D. A., & Wahyudi, M. (2017). Feature selection based on Genetic algorithm, particle swarm optimization and principal component analysis for opinion mining cosmetic product review. *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, 1–6. https://doi.org/10.1109/CITSM.2017.8089278

Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., & Wang, S. (2011). An improved particle swarm optimization for feature selection. *Journal of Bionic Engineering*, *8*(2), 191–200. https://doi.org/10.1016/S1672-6529(11)60020-6

Prasetyaningrum, I., Fathoni, K., & Priyantoro, T. T. J. (2020). Application of recommendation system with AHP method and sentiment analysis. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *18*(3), 1343. https://doi.org/10.12928/telkomnika.v18i3.14778

Que, V. K. S., Iriani, A., & Purnomo, H. D. (2020). Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *9*(2), 162–170. https://doi.org/10.22146/jnteti.v9i2.102

Rana, S., & Singh, A. (2016). Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, *October*, 106–111. https://doi.org/10.1109/NGCT.2016.7877399

Rekha, V., Raksha, R., Patil, P., Swaras, N., & Rajat, G. L. (2019). Sentiment Analysis on Indian Government Schemes Using Twitter data. *2019 International Conference on Data Science and Communication (IconDSC)*, 1–5. https://doi.org/10.1109/IconDSC.2019.8817036

Risawati, R., Ernawati, S., & Maryani, I. (2020). Optimasi Parameter PSO Berbasis SVM untuk Analisis Sentimen Review Jasa Maskapai Penerbangan Berbahasa Inggris. *EVOLUSI : Jurnal Sains Dan Manajemen*, *8*(2), 64–71. https://doi.org/10.31294/evolusi.v8i2.9248

Rustam, F., Ashraf, I., Mehmood, A., Ullah, S., & Choi, G. S. (2019). Tweets Classification on the Base of Sentiments for US Airline Companies. *Entropy*, *21*(11), 1078. https://doi.org/10.3390/e21111078

Sharma, A., & Daniels, A. (2020). *Tweets Sentiment Analysis via Word Embeddings and Machine Learning Techniques*.

Tineges, R., Triayudi, A., & Sholihati, I. D. (2020). Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM). *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *4*(3), 650. https://doi.org/10.30865/mib.v4i3.2181

Wardhani, N. K., Rezkiani, Kurniawan, S., Setiawan, H., Gata, G., Tohari, S., Gata, W., & Wahyudi, M. (2018). Sentiment analysis article news coordinator minister of maritime affairs using algorithm naive bayes and support vector machine with particle swarm optimization. *Journal of Theoretical and Applied Information Technology*, *96*(24), 8365–8378.

Windha Mega, P. D., & Haryoko. (2019). Optimization Of Parameter Support Vector Machine (SVM) using Genetic Algorithm to Review Go-Jek's Services. *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, *6*, 301–304. https://doi.org/10.1109/ICITISEE48480.2019.9003894

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 1, JANUARI, 2022, Pp. 20 – 32
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Analisis Perbandingan Kinerja TCP Vegas dan TCP New Reno Menggunakan Antrian *Drop Tail*

Dony Fahrudy [(1)*], Bambang Sugiantoro [(2)]
Magister Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga, Yogyakarta
e-mail : donnyfahrudi@gmail.com, bambang.sugiantoro@uin-suka.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 5 Juli 2021, direvisi 6 September 2021, diterima 9 September 2021, dan dipublikasikan 25 Januari 2022.

Abstract

TCP was developed to deal with problems that often occur in the network, such as congestion problems. Congestion can occur when the number of packets transmitted in the network approaches the network capacity which can cause network problems. This can be overcome by implementing TCP and queue management. In this research, we will test the performance of TCP Newreno and TCP Vegas using NS-2 in the Drop Tail queue. The performance parameters used are throughput, packet drop, and congestion window with additional buffer capacity. The test results for the congestion window and packet drop parameters, TCP Vegas has better performance when the buffer gets bigger when congestion occurs with the congestion window smaller than TCP New Reno and the average packet drop is 18.33 packets compared to TCP New Reno with an average of 18.33 packets. average 41.67 packets. For throughput parameters, TCP New Reno has better performance with an average of 6.77253 Mbps than TCP Vegas with an average of 4.29693 Mbps. From testing and analysis that TCP Vegas has better performance than TCP New Reno when using Drop Tail queues.

Keywords: Drop Tail, NS-2, TCP Newreno, TCP Vegas, Packet Data Queuing

Abstrak

TCP dikembangkan untuk menangani masalah yang sering terjadi dalam jaringan, seperti masalah *congestion*. *Congestion* dapat terjadi apabila jumlah paket yang ditransmisikan dalam jaringan mendekati kapasitas jaringan yang dapat menyebabkan masalah jaringan. Hal tersebut dapat diatasi dengan menerapkan TCP dan manajemen antrian. Pada penelitian ini akan dilakukan pengujian kinerja TCP Newreno dan TCP Vegas menggunakan NS-2 pada antrian *Drop Tail*. Parameter kinerja yang digunakan adalah *throughput*, *packet drop* dan *congestion window* dengan penambahan kapasitas *buffer*. Hasil pengujian untuk parameter *congestion window* dan *packet drop*, TCP Vegas memiliki kinerja lebih baik ketika *buffer* semakin besar saat terjadi *congestion* dengan *congestion window* lebih kecil dibandingkan TCP New Reno dan rata-rata *packet drop* lebih kecil yaitu 18.33 *packet* dibandingkan TCP New Reno dengan rata-rata 41.67 *packet*. Untuk parameter *throughput*, TCP New Reno memiliki kinerja lebih baik dengan rata-rata 6.77253 Mbps dibandingkan TCP Vegas dengan rata-rata 4.29693 Mbps. Dari pengujian dan analisis bahwa TCP Vegas memiliki kinerja yang lebih baik dari TCP New Reno ketika menggunakan antrian *Drop Tail*.

Kata Kunci: Drop Tail, NS-2, TCP Newreno, TCP Vegas, Antrian Paket Data

1. PENDAHULUAN

TCP/IP (Transmission Control Protocol/Internet Protocol) merupakan standar komunikasi jaringan yang dapat digunakan untuk proses pertukaran data dari satu komputer ke komputer lain dalam jaringan Internet (Syaifuddin et al., 2016). TCP dan IP saling berhubungan, dikarenakan kedua protokol dijadikan satu nama sebab fungsinya yang saling bekerja sama dalam melakukan komunikasi data. Saat ini, telah banyak varian dari TCP, di antaranya TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas dan TCP SACK (Chaudhary & Kumar, 2017).

TCP banyak dikembangkan untuk menangani masalah yang sering terjadi dalam jaringan, seperti masalah kemacetan atau *congestion* (Kembuan, 2016). Kemacetan pada jaringan internet

pertama kali dialami pada akhir tahun 80-an, saat itu menurun drastis sehingga terjadi kolaps atau jatuh, karena pada saat itu belum adanya mekanisme yang dapat menangani hal tersebut. Pada tahun 1986 pertama terjadi *congestion collapse*, kemudian pada tahun 1988 Jacobson mengusulkan teorinya yaitu *Congestion Avoidance and Control* (Fajri, 2016). *Congestion* dapat terjadi apabila jumlah paket yang ditransmisikan sepanjang jaringan mulai mendekati kapasitas jaringan dalam menangani paket-paket, sehingga menyebabkan beberapa masalah yaitu meningkatnya *packet loss*, melambatnya transmisi bahkan dapat menyebabkan kelumpuhan pada jaringan. Oleh karena itu, dibutuhkan sebuah teknik untuk dapat mengontrol kongesti agar tetap mempertahankan jumlah paket data dalam jaringan pada saat kinerja menurun drastis (Kembuan, 2016). Untuk mengatasi hal tersebut dalam penelitian ini akan akan diterapkan TCP dan manajemen antrian yang tepat. TCP Vegas dan TCP New Reno memiliki mekanisme berbeda dalam mengatasi *congestion*, TCP Vegas memiliki sifat proaktif karena dapat menghindari *congestion* sebelum terjadi penumpukan data pada jaringan (Brakmo & Peterson, 1995), sedangkan TCP New Reno memiliki sifat reaktif karena dapat mengendalikan *congestion* ketika terjadi penumpukan data dalam jaringan (Torkey et al., 2012). Untuk manajemen antrian yang digunakan adalah *Drop Tail* yang memiliki mekanisme karakteristik sendiri dalam melayani data pada antrian.

Mekanisme TCP Vegas dalam mengatasi *congestion* lebih kepada *packet delay*. TCP Vegas melihat varian *Round-trip time* (RTT) yang ada dalam mendeteksi *congestion* (Brakmo & Peterson, 1995), RTT merupakan banyaknya waktu yang dibutuhkan oleh suatu paket untuk melakukan perjalanan dari *host* pengirim ke *host* tujuan kemudian kembali lagi ke pengirimnya (Susandi & Pinem, 2014). Jika RTT besar maka jaringan dianggap mengalami *congestion*, sehingga *congestion window* akan dikurangi yang dapat menyebabkan pengiriman data berkurang, jika RTT kecil maka jaringan dianggap dalam keadaan normal, sehingga *congestion window* ditambah yang dapat menyebabkan pengiriman data bertambah (Brakmo & Peterson, 1995). Sedangkan TCP New Reno akan mengirimkan data secara eksponensial dalam jaringan sampai terjadi *packet loss* karena terjadi *congestion*, ketika *packet loss* terdeteksi maka *congestion window* akan dikurangi sampai setengah (Torkey et al., 2012).

Untuk manajemen antrian menggunakan mekanisme antrian *Drop Tail*. Pada antrian *Drop Tail* menggunakan manajemen *First In First Out* (FIFO), dalam hal ini data yang pertama datang akan dilayani, sedangkan apabila kapasitas *buffer* penuh, maka data yang datang akan langsung di drop (Kumar et al., 2012).

Pada penelitian sebelumnya membandingkan kinerja TCP Vegas dan TCP Newreno menggunakan ns-2 dengan penambahan kapasitas *buffer* yang berbeda, penelitian tersebut menggunakan topologi *Abilene* dengan total 31 *node*, didapatkan hasil pengujian *throughput* TCP New Reno memiliki kinerja terbaik ketika menggunakan *Drop Tail* dengan rata-rata 100749.4234kbps sedangkan 94576.815kbps untuk TCP Vegas. Untuk parameter *packet drop*, TCP Vegas memiliki kinerja lebih baik ketika menggunakan antrian *Random Early Detection* dengan rata-rata 0.0002 % dibandingkan dengan TCP New Reno memiliki nilai rata-rata 0.319 %. Dari hasil tersebut dapat disimpulkan bahwa TCP Vegas memiliki kinerja yang lebih baik dari TCP New Reno ketika menggunakan antrian *Random Early Detection* (Pamungkas et al., 2018). Sedangkan pada penelitian lain yang telah dilakukan oleh (Domański et al., 2016), dengan mengevaluasi melalui pendekatan keefektifan kontrol kemacetan TCP New Reno dan TCP Vegas terhadap Perkiraan Aliran Fluida diperoleh hasil bahwa TCP Vegas lebih baik dalam menerapkan algoritma secara adil dibandingkan dengan algoritma TCP Newreno.

Penelitian sebelumnya oleh (Pratama et al., 2015) tentang perbandingan model antrian PCQ, SFQ, RED Dan FIFO Pada Mikrotik Sebagai Upaya Optimalisasi Layanan Jaringan Pada Fakultas Teknik Universitas Tanjungpura didapat hasil bahwa metode antrian *Drop Tail* (FIFO) merupakan metode yang paling optimal untuk di implementasikan pada jaringan internet Fakultas Teknik Universitas Tanjungpura. Penelitian lain juga yang dilakukan oleh (Fajri, 2016) dapat dilihat bahwa *Drop Tail* adalah solusi antrian yang bekerja dengan baik dalam mengatasi antrian

dalam *buffer management* berdasarkan 3 karakteristik yang baik yaitu pada *Packet Dropped*, Pengiriman Ulang, dan *Buffer Usage*.

Dalam penelitian ini akan dilakukan simulasi untuk membandingkan kinerja TCP Vegas dan TCP New Reno dengan menggunakan antrian *Drop Tail* pada NS-2. Simulasi menggunakan topologi *dumbbell* yang diasumsikan sebagai jaringan kabel. Pengujian akan dilakukan dengan skema penambahan kapasitas *buffer* untuk antrian *Drop Tail*. Setelah dilakukan penelitian ini, diharapkan dapat mengetahui varian TCP mana yang memiliki kinerja terbaik ketika dalam jaringan terjadi *congestion* pada antrian *Drop Tail*. Tujuan dari penelitian ini yaitu mengetahui perbandingan kinerja TCP Vegas dan TCP New Reno pada antrian *Drop Tail* dengan melihat nilai dari parameter *throughput*, *packet drop* (*loss*) dan *congestion window*.

2. METODE PENELITIAN

Metode penelitian yang dilakukan merupakan alur dari penelitian yang akan di kerjakan terhadap simulasi jaringan. Penelitian ini dilakukan untuk membangun kerangka algoritma dari masing-masing protokol TCP Vegas dan TCP New Reno. Kerangka penelitian ini dikembangkan dan diimplementasikan pada simulasi yang dijalankan pada *network simulator* NS-2 yang akan digunakan untuk melakukan pengujian dari masing-masing algoritma protokol TCP dengan teknik simulasi topologi *Dumbbell*. Berikut rancangan simulasi jaringan pada penelitian ini.

2.1 Topologi Simulasi

Topologi yang digunakan adalah topologi *dumbbell* dengan 2 node sebagai pengirim yaitu *node* 0 dan *node* 1, 2 *node* sebagai *router* yaitu *node* 2 dan *node* 3, dan 2 *node* sebagai penerima yaitu *node* 4 dan *node* 5. Topologi ini digunakan untuk mengamati kinerja dari TCP Vegas dan TCP New Reno. Berikut rancangan topologi yang digunakan seperti terlihat pada Gambar 1.

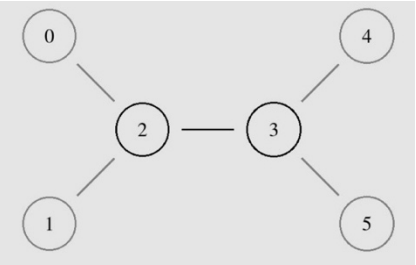
2.2 Parameter Simulasi

Pada penelitian ini ditentukan parameter-parameter jaringan yang bersifat konstan (tetap) yang akan digunakan pada simulasi TCP Vegas dan TCP New Reno menggunakan antrian *Drop Tail*. Berikut parameter-parameter simulasi jaringan yang digunakan seperti pada Tabel 1.

Untuk menentukan *buffer*, peneliti menggunakan perhitungan *Bandwidth Delay Product* pada *link router node* 2 dan *link router node* 3 dengan perhitungan sebagai berikut (PERFSONAR, 2020):

Bandwidth Delay Product = *Bandwidth* * RTT (*delay*)
= 12 Mb * 10 ms
= 12 Mb * 0,01 s
= 0.12 Mbps

Buffer:
1500 *Byte* = 0.012 Mb
0.012 Mb * 8 = 0.096 Mb (<<<<< *bandwidth Delay product*)
0.12 * 10 = 0.12 Mb (===== *bandwidth Delay product*)
0.13 * 12 = 0.144 Mb (>>>>> *bandwidth Delay product*)



Gambar 1 Topologi Simulasi

Tabel 1 Parameter Simulasi

Parameter Simulasi	Nilai
Waktu Simulasi	120 detik
Jumlah Host	4
Link Node n0-n2	Bandwidth: 12Mb Delay: 10ms
Link Node n1-n2	Bandwidth: 12Mb Delay: 10ms
Link Node n2-n3	Bandwidth: 12Mb Delay: 10ms
Link Node n3-n4	Bandwidth: 12Mb Delay: 10ms
Link Node n3-n5	Bandwidth: 12Mb Delay: 10ms
Protocol Transport	TCP Newreno, TCP Vegas
Model Antrian	Drop Tail
Traffic Source	TCP vs TCP
Ukuran Buffer pada Router	8, 10, 12
TCP window size	1000

2.3 TCP New Reno

TCP New Reno menggabungkan berbagai algoritma *congestion control* untuk mengatasi kemacetan diantaranya sebagai berikut (Torkey et al., 2012).

2.3.1 Slow Start and Congestion Avoidance

Pada TCP NewReno, ketika koneksi TCP dimulai, algoritma *Slow Start* menginisialisasi jendela kemacetan (cwnd) ke satu segmen. Pengirim kemudian mulai mengirimkan paket berdasarkan ukuran jendela dan jendela kemacetan bertambah satu segmen hingga cwnd mencapai ambang *slow start* (ssthresh). Pada titik ini, NewReno memasuki fase *Congestion Avoidance* untuk memperlambat laju peningkatan cwnd. Selama penghindaran kemacetan, jendela kemacetan meningkat secara linier satu segmen setiap *round trip time* (RTT) selama kemacetan jaringan tidak terdeteksi (Torkey et al., 2012).

2.3.2 Fast Retransmit and Fast Recovery

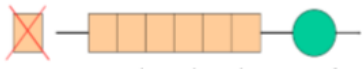
Selama penghindaran kemacetan, berakhirnya penghitung waktu transmisi ulang memberi sinyal kepada pengirim bahwa terjadi kemacetan jaringan. Sehingga pengirim harus memperlambat laju transmisinya. Jika kemacetan ditunjukkan oleh batas waktu, ssthresh diatur ke setengah dari jendela kemacetan saat ini dan jendela kemacetan diatur ke satu segmen dan pengirim masuk ke fase *slow start*. Pengirim masuk ke mode *Fast Retransmit* untuk mentransmisikan ulang paket yang hilang. Kemudian, pengirim menyetel ssthresh ke setengah dari jendela kemacetan dan masuk ke fase *Fast Recovery*. Saat memasuki *Fast Recovery*, pengirim menambah jendela kemacetan satu segmen untuk setiap ACK (*Acknowledgement*) duplikat berikutnya yang diterima. Selama Fast Recovery, TCP NewReno membedakan antara ACK sebagian dan ACK penuh. TCP New Reno berlanjut di *Fast Recovery* sampai semua paket yang beredar selama awal *Fast Recovery* telah diakui. Saat menerima ACK penuh, pengirim menyetel jendela kemacetan (cwnd) ke ssthresh, mengakhiri *Fast Recovery*, dan melanjutkan Penghindaran Kemacetan (Torkey et al., 2012).

2.4 TCP Vegas

TCP Vegas mengontrol ukuran jendelanya dengan mengamati RTT (*round-trip times*) dari paket yang telah dikirim oleh *host* pengirim sebelumnya. Jika RTT yang diamati besar, TCP Vegas mengenali bahwa jaringan mulai padat, dan membatasi ukuran jendela. Jika RTT menjadi kecil, host pengirim TCP Vegas menentukan bahwa jaringan dibebaskan dari kemacetan, dan meningkatkan ukuran jendela lagi. TCP Vegas memiliki fitur lain dalam algoritma *congestion control* yaitu mekanisme *slow-start*. Ukuran jendela bertambah setiap kali paket ACK diterima. Mekanisme *congestion control* yang digunakan oleh TCP Vegas menunjukkan bahwa jika RTT yang diamati dari paket identik, ukuran jendela tetap tidak berubah (Kurata et al., 2000).

2.5 Konfigurasi *Drop Tail*

Drop Tail adalah antrian yang menggunakan algoritma *First In First Out* (FIFO). Mekanismenya yaitu jika ada paket yang pertama kali datang akan dilayani, namun jika *buffer* pada antrian penuh, maka data yang datang akan langsung di *drop* sampai *buffer* memiliki ruang kosong untuk paket yang baru (Kumar et al., 2012). Konfigurasi *Drop Tail* di terapkan pada *node* topologi terhadap *router* dengan *bandwidth* dan *delay* yang telah ditentukan untuk menangani kondisi antrian atau *buffer* dalam jaringan. Berikut ilustrasi *Drop Tail* seperti terlihat pada Gambar 2.

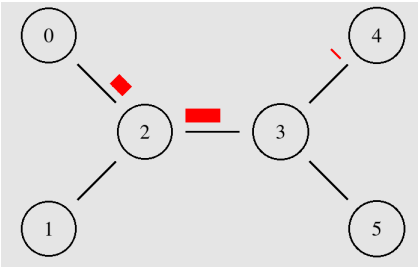


Gambar 2 Ilustrasi *Drop Tail.*

2.6 Skenario Simulasi

2.6.1 Newreno vs Newreno

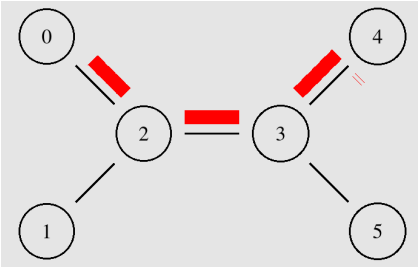
Skenario ini akan menguji congestion control pada TCP Newreno vs TCP Newreno menggunakan topologi *dumbbell* pada antrian *Drop Tail*. Simulasi dimulai dengan menjalankan *traffic* Newreno1 dari n0 menuju n4 dan traffic Newreno2 dari n1 menuju n5 sebagai *traffic* pengganggu. Simulasi ini menggunakan nilai tetap dari *bandwidth* dan *delay* dengan *buffer* = 10. *Traffic* Newreno2 akan mengganggu *flow* jaringan pada detik ke 60. Berikut rancangan topologi pada skenario Newreno vs Newreno seperti terlihat pada Gambar 3.



Gambar 3 Rancangan Topologi Skenario 1

2.6.2 Vegas vs Vegas

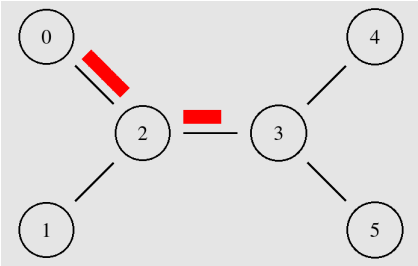
Skenario ini akan menguji *congestion control* pada TCP Vegas vs TCP Vegas menggunakan topologi *dumbbell* pada antrian *Drop Tail*. Simulasi dimulai dengan menjalankan *traffic* Vegas1 dari n0 menuju n4 dan traffic Vegas2 dari n1 menuju n5 sebagai *traffic* pengganggu. Simulasi ini menggunakan nilai tetap dari *bandwidth* dan *delay* dengan buffer = 10. *Traffic* Vegas2 akan mengganggu *flow* jaringan pada detik ke 60. Berikut rancangan topologi pada skenario Vegas vs Vegas seperti terlihat pada Gambar 4.



Gambar 4 Rancangan Topologi Skenario 2

2.6.3 Newreno vs Vegas

Skenario ini akan menguji *congestion control* pada TCP Newreno vs TCP Vegas menggunakan topologi *dumbbell* pada antrian *Drop Tail*. Simulasi dimulai dengan menjalankan *traffic* Newreno dari n0 menuju n4 dan *traffic* Vegas dari n1 menuju n5 sebagai *traffic* pengganggu. Simulasi ini menggunakan nilai tetap dari *bandwidth* dan *delay* dengan buffer yang berbeda yaitu 8, 10, dan 12. *Traffic* Vegas akan mengganggu *flow* jaringan pada detik ke 60. Berikut rancangan topologi pada skenario Newreno vs Vegas seperti terlihat pada Gambar 5.



Gambar 5 Rancangan Topologi Skenario 3

2.7 Parameter Kinerja

2.7.1 Throughput

Throughput merupakan jumlah bit data per satuan waktu yang dikirim atau ditransmisikan ke suatu tujuan melalui jaringan. Semakin besar nilai *throughput* pada TCP maka akan semakin baik. Kualitas TCP dapat terlihat melalui besarnya *throughput* yang dihasilkan. Berikut adalah rumus untuk menghitung *throughput* (Hasanul Fahmi, 2018) seperti terlihat pada Pers. (1).

$$Throughput = \frac{ukuran\ data\ dikirim}{waktu\ pengiriman\ data} \quad (1)$$

2.7.2 Congestion Window

Congestion window (cwnd) adalah satuan dari jumlah paket yang dapat dikirim oleh TCP. *Congestion window* digunakan untuk membatasi jumlah data yang dikirim dalam satu *round trip time* (RTT). Pengirim akan menambah atau mengurangi ukuran *congestion window* terhadap jaringan pada kondisi *congestion* (Taruk & Setyadi, 2016).

2.7.3 Packet Drop

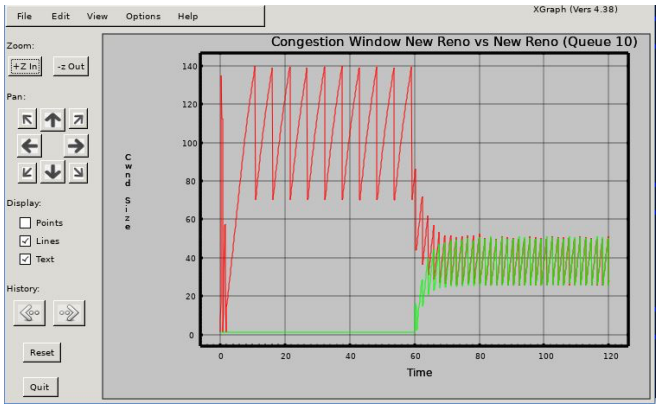
Packet drop adalah paket yang dibuang saat melewati *router*, paket tersebut dibuang dikarenakan *buffer* antrian penuh. Jumlah total paket tersebut akan dibuang dikarenakan paket hilang selama proses transmisi ke tujuan (Orueta et al., 2016). Semakin tinggi jumlah paket yang harus didrop, maka semakin rendah efektifitas penggunaan algoritma untuk paket yang memiliki *deadline* (Taruk & Ashari, 2016).

3. HASIL DAN PEMBAHASAN

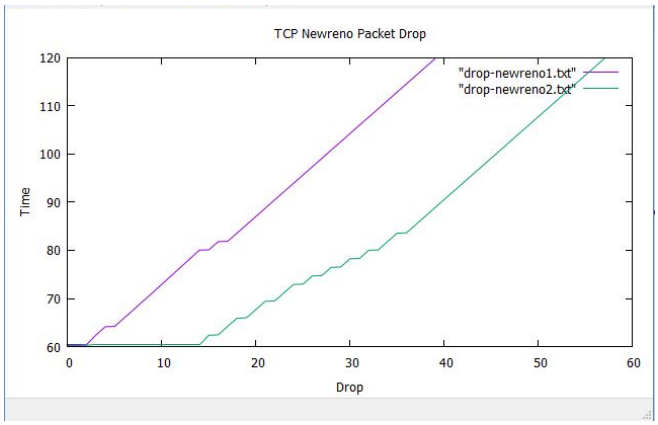
Pengambilan data terhadap pengujian simulasi seperti *congestion window*, nilai *throughput* serta jumlah dan waktu terjadinya *packet drop*, dilakukan sesuai skenario yang telah dirancang sebelumnya. Berikut hasil dari pengambilan data tersebut. Pada simulasi skenario masing-masing TCP, model antrian yang digunakan adalah *Drop Tail*. Pada jenis antrian *Drop Tail*, ketika antrian dalam keadaan penuh, maka paket yang datang akan langsung dibuang hingga terdapat antrian yang kosong.

Pada Gambar 6 dapat dilihat bahwa TCP Newreno1 ditandai dengan garis warna merah dan TCP Newreno2 ditandai dengan garis warna hijau. Pada gambar dapat terlihat bahwa gangguan yang

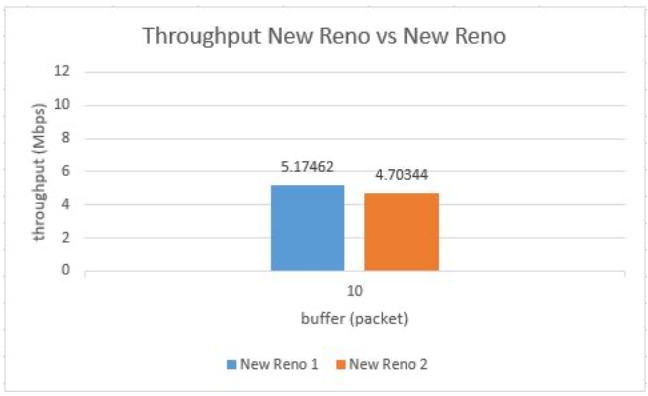
didapat pada TCP Newreno dimulai pada detik ke 60. Setiap paket yang datang dari TCP Newreno1 dan TCP Newreno2 akan ditampung di *buffer* antrian. Ketika antrian penuh maka TCP Newreno1 dan TCP Newreno2 akan dibuang bersamaan. Hal ini akan dilakukan secara berulang sehingga ukuran cwnd TCP Newreno1 sama dengan cwnd TCP Newreno2. Gambar 7 menunjukkan *packet drop* terjadi. Pada detik ke 60.321, TCP Newreno1 mengalami *timeout* karena jumlah *drop* lebih dari 2 berdekatan. Hal ini dikarenakan adanya *traffic* pengganggu TCP Newreno2 seperti yang didukung dengan *congestion window* pada Gambar 6. Selanjutnya, Gambar 8 menunjukkan bahwa *throughput* akan mengalami penurunan ketika adanya gangguan. Penuruan nilai *throughput* akan menjadi setengah dari nilai *bandwidth* yang telah diatur sebelumnya. Hal ini menunjukkan bahwa kedua TCP berbagi *bandwidth* dengan adil.



Gambar 6 *Congestion Window* Newreno vs Newreno

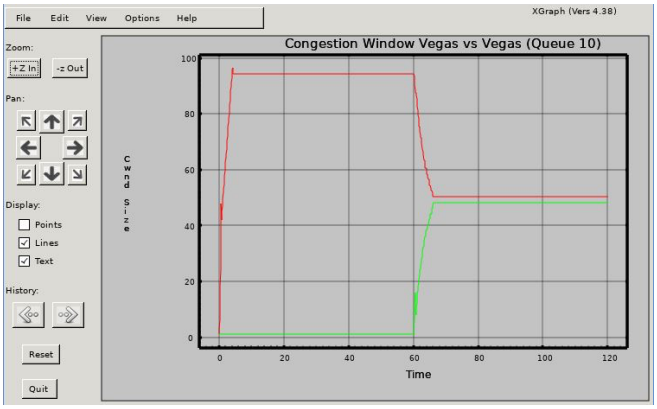


Gambar 7 *Packet Drop* Newreno vs Newreno

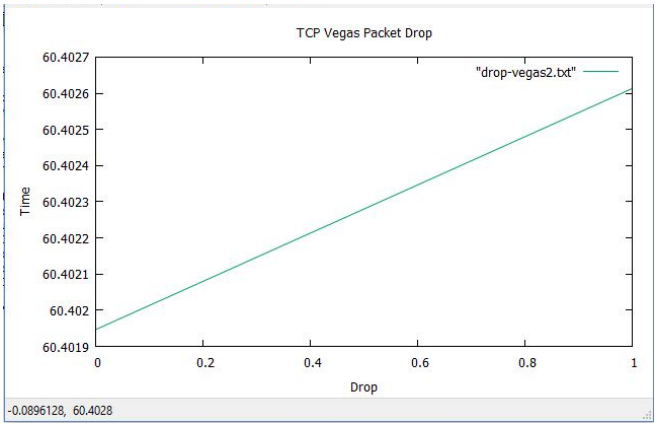


Gambar 8 *Throughput* Newreno vs Newreno

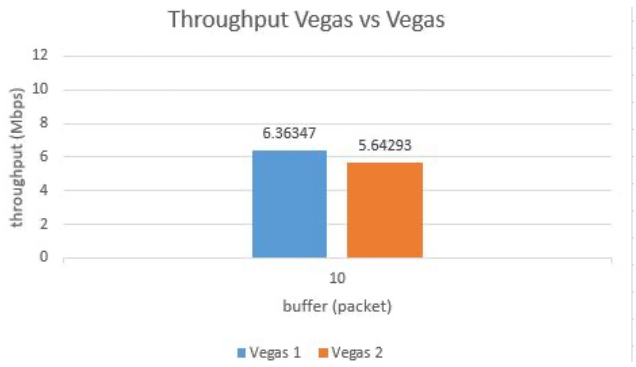
Pada Gambar 9 dapat dilihat bahwa TCP Vegas1 ditandai dengan garis warna merah dan TCP Vegas2 ditandai dengan garis warna hijau. Pada gambar menunjukkan bahwa adanya gangguan yang didapat pada TCP Vegas1 dimulai pada detik ke 60. TCP Vegas1 akan berusaha menurunkan nilai cwnd untuk menghandle adanya gangguan dari TCP Vegas2. Hal ini mengakibatkan adanya *drop* yang dialami oleh TCP Vegas2 pada detik ke 60.40194. Oleh karena itu, nilai cwnd dari TCP Vegas2 akan mendekati nilai cwnd dari TCP Vegas1, sehingga nilai dari kedua TCP seimbang. Selanjutnya, Gambar 11 menunjukkan bahwa *throughput* akan mengalami penurunan ketika adanya gangguan. Penuruan nilai *throughput* akan menjadi setengah dari *bandwidth* yang telah di atur sebelumnya. Hal ini menunjukkan bahwa kedua TCP berbagi *bandwidth* dengan adil.



Gambar 9 *Congestion Window* Vegas vs Vegas

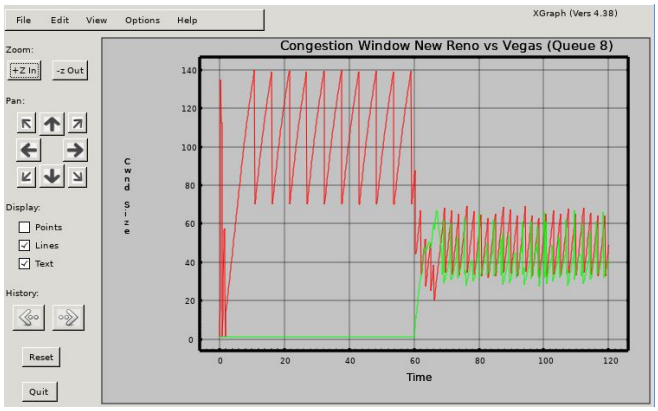


Gambar 10 *Packet Drop* Vegas vs Vegas

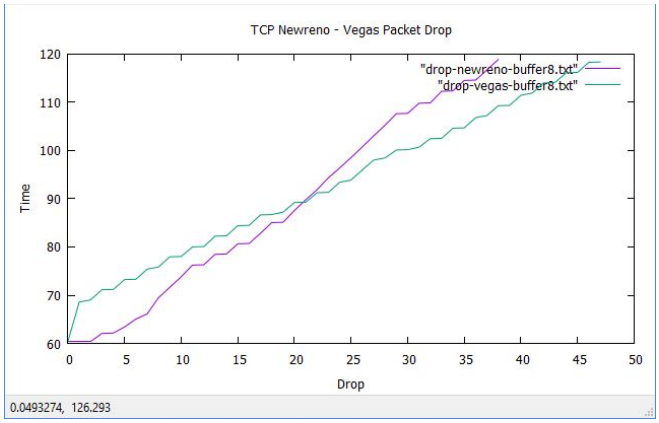


Gambar 11 *Throughput* Vegas vs Vegas

Pada Gambar 12 dapat dilihat bahwa TCP Newreno ditandai dengan garis warna merah dan TCP Vegas ditandai dengan garis warna hijau. Pada gambar menunjukkan bahwa *congestion window* dari TCP Newreno lebih kecil dari TCP Vegas. Hal ini dikarenakan cara kerja dari TCP Newreno dipengaruhi oleh adanya *packet loss* yang menyebabkan *congestion window* turun. Hal berbeda dialami oleh TCP Vegas. Pada TCP Vegas cara kerjanya dipengaruhi oleh variasi *delay* yang akan menyebabkan penurunan *congestion window*. Semakin kecil antrian *buffer* maka semakin kecil variasi *delay* yang dialami oleh TCP Vegas. Hal inilah yang menyebabkan *congestion window* TCP Vegas lebih besar dari TCP Newreno, hal tersebut didukung dengan banyaknya packet yang di *drop* pada TCP Vegas ketika jaringan mengalami *congestion* seperti pada Gambar 13.



Gambar 12 *Congestion Window* Newreno vs Vegas, *buffer* 8

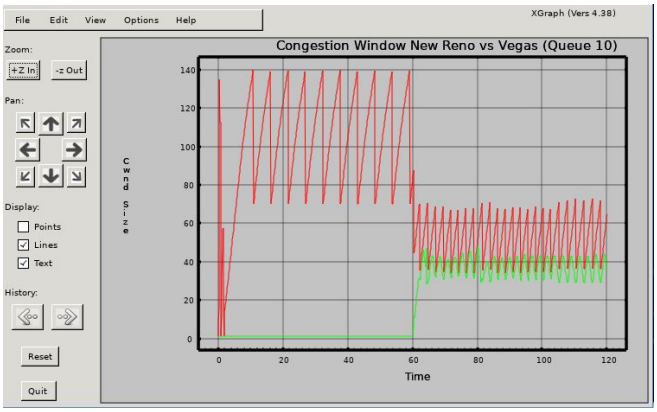


Gambar 13 *Packet Drop* Newreno vs Vegas, *buffer* 8

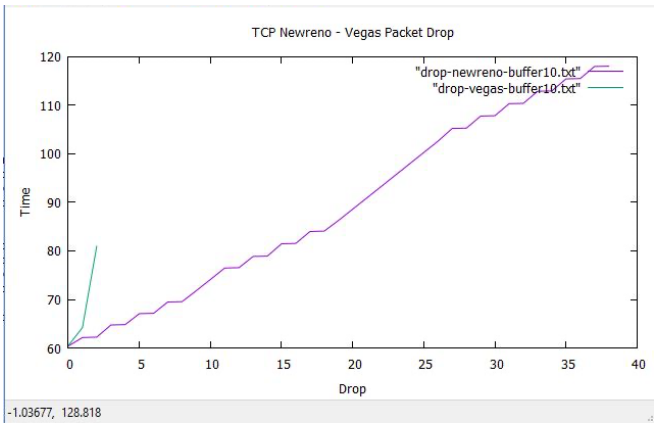
Pada Gambar 14 dapat dilihat bahwa TCP Newreno ditandai dengan garis warna merah dan TCP Vegas ditandai dengan garis warna hijau. Pada gambar menunjukkan bahwa *congestion window* dari TCP Newreno lebih besar dari TCP Vegas. Hal ini dikarenakan TCP New Reno dalam mengirimkan paket data tidak memperhatikan kondisi jaringan dengan mengirimkan paket data secara cepat, sehingga *congestion window* meningkat secara eksponential, namun ketika jaringan mengalami *congestion*, akan lebih banyak terjadi *packet drop* seperti pada Gambar 15. Hal tersebut didukung cara kerja dari TCP Newreno yang dipengaruhi oleh adanya *packet loss*. Sedangkan pada TCP Vegas semakin besar *buffer* antrian maka variasi *delay* juga semakin besar yang menyebabkan *congestion window* semakin kecil, sehingga akan lebih sedikit terjadi *packet drop* dikarenakan *congestion* yang dialami oleh jaringan lebih kecil.

Pada Gambar 16 dapat dilihat bahwa TCP Newreno ditandai dengan garis warna merah dan TCP Vegas ditandai dengan garis warna hijau. Pada gambar menunjukkan bahwa *congestion window* dari TCP Newreno lebih besar dari TCP Vegas. Hal ini dikarenakan TCP New Reno dalam

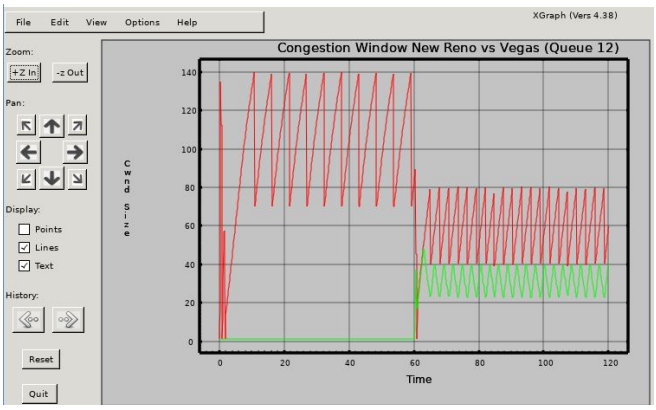
mengirimkan paket data tidak memperhatikan kondisi jaringan dengan mengirimkan paket data secara cepat, sehingga *congestion window* meningkat secara eksponential, namun ketika jaringan mengalami *congestion*, akan lebih banyak terjadi *packet drop* seperti pada Gambar 17. Hal tersebut didukung cara kerja dari TCP Newreno yang dipengaruhi oleh adanya packet loss. Sedangkan pada TCP Vegas semakin besar *buffer* antrian maka variasi *delay* juga semakin besar yang menyebabkan *congestion window* semakin kecil, sehingga akan lebih sedikit terjadi *packet drop* dikarenakan *congestion* yang dialami oleh jaringan lebih kecil.



Gambar 14 *Congestion Window* Newreno vs Vegas, buffer 10

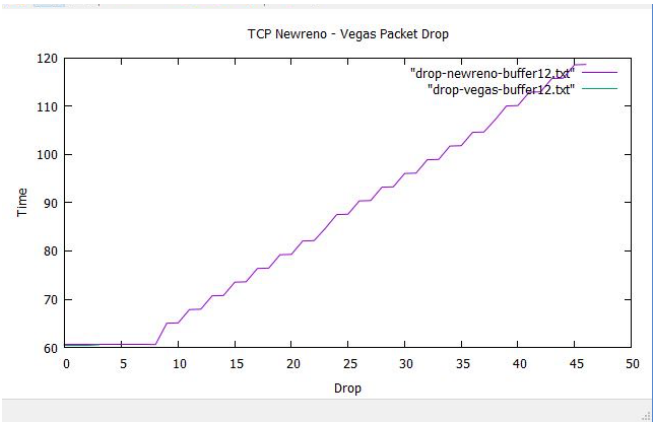


Gambar 15 *Packet Drop* Newreno vs Vegas, buffer 10

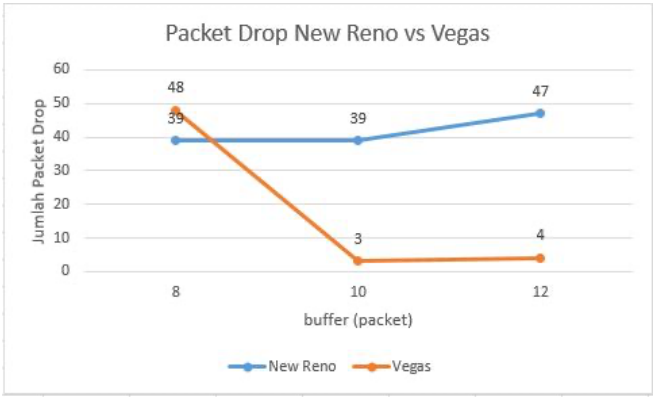


Gambar 16 *Congestion Window* Newreno vs Vegas, buffer 12

Berdasarkan *graph packet drop* pada Gambar 13, 15 dan 17, pada Gambar 18 merupakan jumlah *packet drop* pada masing-masing TCP untuk *buffer* 8, 10 dan 12 yang menujukkan bahwa *packet drop* TCP Newreno terlihat mengalami kenaikan ketika antrian *buffer* semakin besar dengan rata-rata lebih besar yaitu 41.67 packet, hal tersebut dikarenakan TCP New Reno dalam mengirimkan data tidak memperhatikan kondisi jaringan dengan mengirimkan data secara cepat, sehingga *congestion window* meningkat secara eksponential yang menyebabkan terjadinya *packet drop*. Sedangkan pada TCP Vegas terlihat *packet drop* mengalami penurunan drastis ketika antrian *buffer* semakin besar dengan rata-rata lebih kecil yaitu 18.33 packet, hal tersebut dikarenakan TCP Vegas dalam mengirimkan data akan melihat kondisi jaringan terlebih dahulu dengan menghitung varian RTT, ketika antrian *buffer* semakin besar dan paket data yang dapat ditampung semakin banyak, maka nilai RTT menjadi besar dan membuat TCP Vegas mengurangi jumlah data yang dikirim, sehingga besar data yang dapat di tampung pada *buffer* tidak terlalu banyak yang akan membuat *packet drop* menjadi kecil.



Gambar 17 *Packet Drop* Newreno vs Vegas, buffer 12

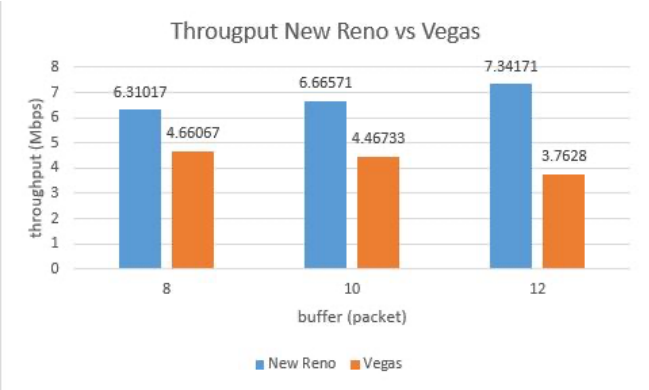


Gambar 18 *Packet Drop* Newreno vs Vegas

Gambar 19 menunjukkan bahwa nilai *throughput* TCP Newreno lebih tinggi daripada TCP Vegas baik ketika antrian *buffer* semakin besar dengan rata-rata lebih besar yaitu 6.77253 Mbps. Semakin tinggi *buffer* maka akan semakin tinggi nilai *throughput* TCP Newreno. TCP Newreno akan mengirim paket sebanyak mungkin sehingga semakin banyak data yang dilayani dan meski demikian *packet drop* terjadi, TCP New Reno akan melanjutkan ke fase *fast recovery* untuk menjaga *throughput* tetap tinggi.

Nilai *throughput* dari TCP Vegas lebih kecil dari TCP Newreno ketika *buffer* antrian besar dengan rata-rata lebih kecil yaitu 4.29693 Mbps. Hal ini dikarenakan TCP Vegas dalam mengirimkan data akan melihat kondisi jaringan berdasarkan varian RTT, jika RTT besar maka TCP Vegas akan mengurangi data yang dikirim, jika RTT kecil maka TCP Vegas akan mengirimkan data lebih

banyak. TCP Vegas mengalami penurunan ketika penambahan kapasitas *buffer*, hal tersebut disebabkan semakin besar kapasitas *buffer* maka data yang ditampung lebih banyak sehingga nilai RTT menjadi besar, saat nilai RTT besar TCP Vegas menganggap jaringan dalam keadaan *congestion* sehingga data yang dikirim semakin sedikit yang menyebabkan nilai *throughput* nya menjadi kecil.



Gambar 19 *Throughput* Newreno vs Vegas

4. KESIMPULAN

Dari hasil analisa dan pengujian yang dilakukan, dapat disimpulkan bahwa penerapan TCP Vegas dan TCP New Reno pada antrian *Drop Tail* dengan parameter uji yang berbeda telah berhasil dilakukan. Penerapan dilakukan dengan cara melakukan instalasi NS-2, setelah instalasi NS-2 berhasil dilanjutkan dengan konfigurasi *script* simulasi. Pengujian dilakukan dengan skema penambahan kapasitas *buffer* untuk antrian *Drop Tail*. Kemudian melakukan uji coba pada TCP Vegas dan TCP New Reno dengan melihat nilai dari parameter *throughput*, *packet drop* dan *congestion window*.

Hasil yang didapat dari pengujian yaitu untuk parameter *congestion window*, TCP Vegas memiliki kinerja lebih baik ketika *buffer* semakin besar. Pada antrian lebih kecil yaitu *buffer* 8, *congestion window* pada TCP Vegas lebih besar dibandingkan TCP New Reno namun tidak ada perbedaan signifikan, sedangkan pada *buffer* 10 dan 12, *congestion window* pada TCP Vegas lebih kecil dibandingkan TCP New Reno. Untuk parameter *packet drop*, TCP Vegas memiliki kinerja lebih baik ketika *buffer* semakin besar dengan rata-rata lebih kecil yaitu 18.33 *packet* dibandingkan dengan TCP New Reno memiliki nilai rata-rata yang lebih besar yaitu 41.67 *packet*. Untuk parameter *throughput*, TCP New Reno memiliki kinerja lebih baik dengan rata-rata lebih besar yaitu 6.77253 Mbps dibandingkan dengan TCP Vegas dengan rata-rata yang lebih kecil yaitu 4.29693 Mbps. Oleh karena itu, dari hasil pengujian dan analisis yang telah dilakukan pada TCP Vegas dan TCP Newreno dapat disimpulkan bahwa TCP Vegas memiliki kinerja yang lebih baik dari TCP New Reno ketika menggunakan antrian *Drop Tail* yaitu pada parameter kinerja *congestion window* dan *packet drop*.

Adapun saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah membandingkan varian TCP yang lainnya yang memiliki algoritma *congestion window* yang sama. Serta melakukan pengujian dengan mekanisme antrian lainnya.

DAFTAR PUSTAKA

Brakmo, L. S., & Peterson, L. L. (1995). TCP Vegas: end to end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, *13*(8), 1465–1480. https://doi.org/10.1109/49.464716

Chaudhary, P., & Kumar, S. (2017). A Review of Comparative Analysis of TCP Variants for Congestion Control in Network. *International Journal of Computer Applications*, *160*(8), 28–

34. https://doi.org/10.5120/ijca2017913087

Domański, A., Domańska, J., Pagano, M., & Czachórski, T. (2016). The Fluid Flow Approximation of the TCP Vegas and Reno Congestion Control Mechanism. In *Communications in Computer and Information Science* (Vol. 659, pp. 193–200). https://doi.org/10.1007/978-3-319-47217-1_21

Fajri, M. (2016). Simulasi Antrian Paket Data Jaringan dengan Mekanisme Drop Tail. *Jurnal Ilmiah FIFO*, *8*(2), 151. https://doi.org/10.22441/fifo.v8i2.1310

Hasanul Fahmi. (2018). Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik. *Jurnal Teknologi Informasi Dan Komunikasi*, *7*(2), 98–105.

Kembuan, O. (2016). Analisa Packet Loss Transmission Control Protocol (TCP) RENO pada Jaringan Intranet Menggunakan NS2 (Network Simulator). *Engineering Education Journal (E2J-UNIMA)*, *4*(3), 24.

Kumar, A., Sharma, A. K., & Singh, A. (2012). Comparison and Analysis of Drop Tail and RED Queuing Methodology in PIM-DM Multicasting Network. *International Journal of Computer Science and Information Technologies*, *3*(2), 3816–3820.

Kurata, K., Hasegawa, G., & Murata, M. (2000). Fairness comparisons between TCP Reno and TCP Vegas for future deployment of TCP Vegas. *INET Conferences*, 1–20.

Orueta, G. D., Ruiz, E. S. C., Alonso, N. O., & Gil, M. C. (2016). Quality of Service. In *Ad Hoc Mobile Wireless Networks* (pp. 200–221). CRC Press. https://doi.org/10.1201/b13094-7

Pamungkas, G. W., Yahya, W., & Nurwarsito, H. (2018). Analisis Perbandingan Kinerja TCP Vegas Dan TCP New Reno Menggunakan Antrian Random Early Detection Dan Droptail. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK)*, *2*(10), 3239–3248.

PERFSONAR. (2020). *Lab 6 : Bandwidth-delay Product and TCP Buffer Size*.

Pratama, T., Irwansyah, M. A., & Yulianti. (2015). Perbandingan Metode PCQ, SFQ, RED Dan FIFO Pada Mikrotik Sebagai Upaya Optimalisasi Layanan Jaringan Pada Fakultas Teknik Universitas Tanjungpura. *Jurnal Sistem Dan Teknologi Informasi*, *3*(3), 298–303.

Susandi, H., & Pinem, M. (2014). Analisis Kualitas Layanan Data pada Jaringan Telekomunikasi Berbasis CDMA EVDO Rev . A. *Jurnal Singuda Ensikom*, *6*(2), 93–98.

Syaifuddin, M., Andika, B., & Ginting, R. I. (2016). Analisis Celah Keamanan Protocol TCP/IP. *Jurnal Ilmiah Sains Dan Teknologi (SAINTIKOM)*, *16*(2), 130–135.

Taruk, M., & Ashari, A. (2016). Analisis Throughput Varian TCP Pada Model Jaringan WiMAX. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *10*(2), 115. https://doi.org/10.22146/ijccs.15529

Taruk, M., & Setyadi, H. J. (2016). Analisis Mekanisme Penanganan Kemacetan (Congestion Control) pada Algoritma Varian. *Konferensi Nasional Ilmu Komputer (KONIK)*, 1–4.

Torkey, H., Attiya, G., & Z. Morsi, I. (2012). Modified Fast Recovery Algorithm for Performance Enhancement of TCP-NewReno. *International Journal of Computer Applications*, *40*(12), 30–35. https://doi.org/10.5120/5018-7351

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 1, JANUARI, 2022, Pp. 33 – 45
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Pengamanan Citra Digital Berbasis Kriptografi Menggunakan Algoritma Vigenere Cipher

Imam Riadi [(1)], Abdul Fadlil [(2)], Fahmi Auliya Tsani [(3)*]
[1] Sistem Informasi, Fakultas Sains dan Teknologi Terapan, Universitas Ahmad Dahlan, Yogyakarta
[2] Teknik Elektro, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Yogyakarta
[3] Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Yogyakarta
e-mail : imam.riadi@is.uad.ac.id, fadlil@mti.uad.ac.id, fahmi1807048017@webmail.uad.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 15 Juli 2021, direvisi 15 Desember 2021, diterima 15 Desember 2021, dan dipublikasikan 25 Januari 2022.

Abstract

Cryptography is one of the most popular methods in data security by making data very difficult to read or even unreadable. One of the well-known techniques or algorithms in cryptography is Vigenere Cipher. This classic algorithm is classified as a polyalphabetic substitution cipher-based algorithm. Therefore, this algorithm tends to only handle data in text form. By this research, a console-based application has been developed which is made from PHP programming language to be able to encrypt and decrypt digital image media using Vigenere Cipher. The encryption process is done by first converting a digital image into a base64 encoding format so that the encryption process can be carried out using the tabula recta containing the radix-64 letter arrangement used for base64 encoding. Conversely, the decryption process is carried out by restoring the encrypted file using radix-64 letters, so we get the image file in the base64 encoding format. Then, the image with the base64 encoding format is decoded into the original file. The encryption process took less than 0,2 seconds and 0.19 seconds for the decryption process and 33.34% for average file size addition on the encrypted file from the original file size. Testing on ten different images with different sizes and dimensions showed a 100% success rate which means this research was successfully carried out.

Keywords: Digital Image Encryption, Vigenere Cipher, Classical Cryptography, Base64 Encoding, Data Security

Abstrak

Kriptografi merupakan salah satu metode populer dalam pengamanan data dengan cara membuat data sulit atau bahkan tidak bisa dibaca. Salah satu teknik atau algoritma yang terkenal dalam kriptografi adalah Vigenere Cipher. Algoritma kriptografi klasik ini termasuk dalam kategori algoritma berbasis *polyalphabetic substitution cipher*. Oleh karena itu, algoritma ini cenderung hanya bisa menangani data dalam bentuk teks. Pada penelitian ini, dikembangkan aplikasi berbasis konsol yang dibuat dalam bahasa pemrograman PHP dengan tujuan agar bisa melakukan enkripsi dan dekripsi pada media citra digital menggunakan Vigenere Cipher. Proses enkripsi dilakukan dengan terlebih dahulu mengubah suatu citra digital menjadi format *encoding* base64, sehingga bisa dilakukan proses enkripsi dengan memakai tabula recta yang berisi susunan huruf radix-64 yang digunakan untuk proses encoding base64. Sebaliknya, proses dekripsi dilakukan dengan mengembalikan *file* yang sudah dienkripsi menggunakan susunan huruf radix-64, sehingga didapatkan *file* citra dalam format *encoding* base64. Lalu, citra berformat *encoding* base64 ini di-*decode* menjadi *file* asli. Proses enkripsi membutuhkan waktu kurang dari 0,2 detik dan 0,19 detik untuk proses dekripsi serta mengalami penambahan ukuran rata-rata sebesar 33,34% pada *file* hasil enkripsi dari ukuran *file* semula. Pengujian pada sepuluh citra digital dengan ukuran dan dimensi berbeda-beda menunjukkan tingkat keberhasilan sebesar 100%, hal ini berarti bahwa penelitian berhasil dilakukan.

Kata Kunci: Enkripsi Citra Digital, Vigenere Cipher, Kriptografi Klasik, Pengkodean Base64, Keamanan Data

1. PENDAHULUAN

Keamanan atau *security* merupakan salah satu aspek penting yang seharusnya dipenuhi dari suatu informasi atau data. Keamanan sangat penting karena berhubungan dengan data sensitif dengan cara melindungi dari akses yang tidak sah, pengubahan, maupun penghapusan (Awad et al., 2019). Ada beberapa aspek dalam keamanan data antara lain *authentication*, *confidentiality/privacy*, *integrity*, dan *non-repudiation*. Beberapa poin ini bisa diselesaikan dengan menggunakan teknik kriptografi (Munir, 2006). Hermansa et al. (2019) mendefinisikan bahwa kriptografi merupakan teknik yang digunakan untuk mengamankan suatu data melalui proses enkripsi sehingga data menjadi sulit dibaca atau dibuka oleh seseorang yang tidak berwenang karena tidak memiliki kunci untuk melakukan dekripsi. Dengan kata lain, kriptografi mampu mengubah isi suatu data menjadi data lain yang acak (Fadlil et al., 2020b).

Secara garis besar, kriptografi dibedakan menjadi dua yaitu kriptografi klasik dan kriptografi modern. Salah satu algoritma atau teknik dalam kriptografi klasik yang populer adalah Vigenere Cipher. Algoritma ini mengimplementasikan teknik substitusi yaitu proses penyandian dengan mengubah isi suatu data berdasarkan kunci yang digunakan agar tidak terbaca maknanya (Setiadi et al., 2018). Vigenere Cipher menggunakan bujur sangkar Vigenere dalam melakukan proses enkripsi maupun dekripsi, sehingga membuat algoritma ini dikenal mudah dipahami dan diimplementasikan (Munir, 2006). Contoh bujur sangkar Vigenere bisa dilihat pada Gambar 1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 1 Contoh Tabel Bujur Sangkar Vigenere

Saat ini, kriptografi tidak hanya digunakan untuk data-data berjenis teks saja namun juga sangat memungkinkan untuk diaplikasikan pada jenis data yang lain seperti citra, video, dan suara (Sinaga et al., 2018). Suatu algoritma kriptografi bisa dikatakan bagus jika mampu mempertahankan aspek kerahasiaan dari pesan yang dienkripsi dan tidak mudah dipecahkan oleh orang yang tidak berhak untuk mengakses data tersebut (Anwar et al., 2019).

Beberapa penelitian terdahulu sudah banyak yang mengangkat tema pengaplikasian Vigenere Cipher ini untuk mengamankan berbagai jenis data. (Gerhana et al., 2016) dalam penelitiannya

mengangkat tema pengaplikasian Vigenere Cipher pada media citra digital dengan melakukan substitusi kode warna pada setiap pikselnya berdasarkan kunci yang dimasukkan. Sebagai hasilnya, terbentuk citra lain dengan warna yang acak.

Gunadhi & Sudrajat (2016) mengimplementasikan Vigenere Cipher yang sudah dimodifikasi untuk melakukan pengamanan pada data rekam medis pasien, sehingga menjadikan data rekam medis pasien lebih aman dari serangan para *cryptanalyst*.

Penelitian lain dilakukan oleh (Mandal & Deepti, 2016) dengan mengimplementasikan skema enkripsi *multi level*. Metode yang digunakan yaitu dengan menggunakan kunci yang memiliki panjang karakter sama dengan *plain text* sehingga menghasilkan *cipher text* pertama. Tidak berhenti sampai di sini, *cipher text* pertama ini lalu dienkripsi lagi dengan kunci yang sama dengan *cipher text* pertama sehingga dihasilkan *cipher text* kedua. Sebagai kesimpulannya, dibandingkan dengan beberapa algoritma kriptografi lainnya (AES, Blowfish, dan RC5) metode ini memiliki hasil yang sulit dipecahkan oleh *cryptanalyst* dan juga memiliki kompleksitas komputasi yang lebih rendah sehingga cocok digunakan untuk aplikasi yang ringan dan memiliki *resource* yang terbatas.

Soofi et al. (2016) mencoba melakukan sedikit modifikasi pada tabel bujur sangkar Vigenere dengan mengganti urutan setiap karakternya dan menambahkan satu karakter "&" sebagai pengganti karakter spasi (*white space*). Dengan metode ini, dihasilkan algoritma Vigenere yang lebih kuat terhadap serangan dengan metode Kasiski dan Friedman.

Beberapa penelitian mengenai Vigenere Cipher dilakukan dengan menggabungkan Vigenere Cipher dengan teknik-teknik lain. Nasution et al. (2017) menggabungkan Vigenere Cipher dengan teknik kompresi Goldbach Codes. Hasil penggabungan ini membuahkan *cipher text* yang susah diprediksi meskipun menggunakan serangan metode Kasiski, hal ini dikarenakan kumpulan karakter yang dihasilkan berbeda dengan karakter yang digunakan pada *plain text*.

Maruf et al. (2015) melakukan penelitian dengan menggabungkan Vigenere Cipher dengan XTEA (*Extended Tiny Encryption Algorithm*) *block cipher*. XTEA dipilih karena sudah teruji kekuatannya dalam mengamankan data berjenis teks. Penggabungan dua teknik ini diberi nama VixTEA dan terbukti meningkatkan keamanan berdasarkan beberapa pengujian yang dilakukan yaitu analisis frekuensi, analisis nilai entropi, dan analisis serangan *brute force*. Hasil pengujian bahkan juga menunjukkan bahwa konsep ini tidak mempengaruhi performa dari algoritma itu sendiri.

Rojali et al. (2016) dalam penelitiannya menggabungkan beberapa metode sebagai langkah untuk mengamankan data. Beberapa metode yang digunakan yaitu enkripsi, pembangkitan kunci, kompresi data, dan steganografi. Enkripsi yang digunakan yaitu Vigenere Cipher yang sudah dimodifikasi menggunakan komposisi bujur sangkar Vigenere sesuai dengan susunan huruf, angka, dan simbol yang ada pada *keyboard*. Sedangkan *key* yang digunakan dibangkitkan melalui *chaos function*. Proses selanjutnya yaitu melakukan kompresi pada data yang sudah dienkripsi menggunakan *Dictionary Based Compression*. Sebagai Langkah terakhir, data yang telah dikompres disembunyikan ke dalam suatu citra digital menggunakan steganografi dengan metode *Least Significant Bit* (LSB).

Masih dengan media *plain text*, Saputra et al. (2017) mengimplementasikan Vigenere Cipher dengan memanfaatkan citra *grayscale* berukuran 5 x 5 piksel sebagai kunci. Kunci citra *grayscale* ini dikonversi menjadi karakter ASCII, sehingga menjadi susunan karakter yang bisa diolah ke dalam Vigenere Cipher.

Subandi et al. (2018) dalam penelitiannya melakukan enkripsi citra digital dengan melakukan dua kali proses enkripsi menggunakan Vigenere Cipher dan mengadopsi *expansion key* menggunakan algoritma RC6 pada media teks. Penelitian ini memiliki tujuan untuk mengetahui perbandingan perbedaan ukuran suatu data sebelum dan sesudah dienkripsi (*avalanche effect*)

pada beberapa skenario seperti penggunaan Vigenere Cipher secara standar, penggabungan dengan *expansion key* RC6, dan lain-lain.

Serupa dengan penelitian dari Subandi et al. (2018) yang bertujuan untuk mengetahui perbandingan *avalanche effect*, Rihartanto et al. (2020) menggunakan Vigenere Cipher yang sudah dimodifikasi dengan cara memperluas jangkauan karakter yang bisa diakomodasi menjadi 128 buah sesuai dengan jumlah karakter ASCII standar serta melakukan rotasi matriks bujur sangkar. Implementasi dari proses tersebut menghasilkan nilai *avalanche effect* sekitar 45% hingga 49%.

Fadlil et al. (2020b) melakukan pendekatan yang berbeda pada penelitiannya mengenai implementasi Vigenere Cipher yaitu dengan mengombinasikan Jaringan Syaraf Tiruan (JST) dengan Vigenere Cipher. Penelitian ini menggunakan JST sebagai generator kunci dengan memasukkan parameter *hidden neurons* (K), *input neurons* (N), dan bobot (L) sehingga dihasilkan karakter acak yang bisa digunakan dalam proses enkripsi dan dekripsinya. Melalui pendekatan ini diklaim memiliki sedikit kemungkinan memunculkan kunci yang sama meskipun memasukkan nilai parameter yang sama berulang kali.

Prabowo & Hangga (2015) juga mengimplementasikan metode pembangkitan kunci untuk digunakan di Vigenere Cipher. Caesar Cipher dipilih sebagai pembangkit kunci, dengan kata lain parameter kunci yang dimasukkan oleh pengguna akan dienkripsi terlebih dahulu menggunakan Caesar Cipher, sehingga dihasilkan *encrypted key* yang akan digunakan untuk proses enkripsi-dekripsi menggunakan Vigenere Cipher. Metode ini dipilih karena dari contoh yang digunakan memperlihatkan bahwa Vigenere Cipher dalam kondisi standar berpotensi menghasilkan pengulangan kata saat menggunakan kunci berulang dengan panjang kunci yang cukup pendek. Sebagai hasil dari penelitian ini, disimpulkan bahwa metode yang digunakan mampu mengurangi potensi pengulangan kata bahkan tidak ditemukan adanya pengulangan kata.

Hernawandra et al. (2018) melibatkan citra digital dalam penelitiannya dalam mengamankan data berupa teks dengan terlebih dahulu melakukan proses enkripsi menggunakan Vigenere Cipher dan substitusi. *Cipher text* yang dihasilkan dari proses enkripsi kemudian disembunyikan pada media citra digital dengan menggunakan teknik steganografi LSB 4 bit. Keluaran dari penelitian ini berupa aplikasi yang berjalan pada platform Android. Penelitian ini menghasilkan kesimpulan bahwa aplikasi yang dibangun mampu mengamankan pesan melalui metode steganografi LSB 4 bit yang digabungkan dengan enkripsi substitusi dan Vigenre Cipher serta memiliki *avalanche effect* rata-rata sebesar 12,77%.

Penelitian mengenai kriptografi menggunakan algoritma Vigenere Cipher yang telah dilakukan sebelumnya banyak yang berfokus pada pengamanan data jenis teks. Penelitian yang berfokus pada data jenis citra digital masih sangat jarang ditemukan. Padahal, citra digital merupakan salah satu jenis media yang sangat populer digunakan untuk berkomunikasi baik secara daring maupun langsung (Zebua & Ndruru, 2017). Oleh sebab itu, penelitian ini akan mengembangkan pengamanan data jenis citra digital menggunakan metode Vigenere Cipher. Pembuktian validitas hasil penelitian dengan membandingkan nilai *hash file* asli dengan nilai *hash file* hasil dekripsi. Penelitian ini juga akan menyajikan data mengenai waktu yang dibutuhkan untuk proses enkripsi dan dekripsi serta menghitung persentase rata-rata perubahan ukuran *file* yang dihasilkan dengan cara membandingkan ukuran *file* asli dengan ukuran *file* setelah dienkripsi. Hasil pengujian akan dibandingkan dengan penelitian sejenis yang menggunakan algoritma Arnold's Cat Map seperti yang dilakukan oleh (Rachmawanto, dkk., 2019) dan algoritma Elgamal dengan mode Electronic Code Book (ECB) seperti yang dilakukan oleh (Rizal, dkk., 2016). Penelitian ini diharapkan dapat memberikan wawasan mengenai pentingnya pengamanan data atau *file*, terutama data jenis citra digital.

2. METODE PENELITIAN

Data atau sistem yang tidak aman tentu akan berdampak buruk (Riadi et al., 2020). Salah satu metode yang dapat digunakan untuk mempertahankan kerahasiaan suatu data adalah dengan

mengubahnya menjadi data tersandi yang tidak bermakna, proses ini bisa biasa disebut sebagai kriptografi (Yunita et al., 2019). Menurut terminologinya, kriptografi merupakan ilmu dan seni yang digunakan untuk mengamankan pesan ketika pesan dikirim dari suatu sumber ke tempat tujuan. Proses ini terdiri dari tiga fungsi dasar antara lain (Ariyus, 2006):

a. Enkripsi, proses mengubah pesan asli menjadi berbentuk kode-kode yang susah atau bahkan tidak bisa dimengerti.
b. Dekripsi, proses kebalikan dari enkripsi yaitu mengubah pesan yang sudah terenkripsi menjadi pesan asli.
c. Kunci, sekumpulan parameter yang digunakan dalam proses enkripsi maupun dekripsi.

Schneier dan Menezes (seperti yang diacu dalam Munir, 2006) menerangkan bahwa kriptografi memiliki beberapa tujuan pada beberapa aspek keamanan sebagai berikut:

a. Kerahasiaan (*confidentiality*), bertujuan agar pesan tidak bisa dibaca oleh pihak-pihak yang tidak berhak.
b. Integritas data (*data integrity*), bertujuan agar mendapat jaminan bahwa pesan masih asli/utuh dan tidak dimanipulasi saat pengiriman.
c. Otentikasi (*authentication*), bertujuan untuk mengidentifikasi kebenaran pihak-pihak yang saling berkomunikasi maupun mengidentifikasi kebenaran pesan.
d. Nirpenyangkalan (*non-repudiation*), bertujuan agar tidak ada penyangkalan oleh pihak-pihak yang berkomunikasi.

Penelitian ini mengimplementasikan algoritma Vigenere Cipher yang sudah dimodifikasi dan menghasilkan *output* berupa aplikasi berbasis *console*. Modifikasi yang dilakukan berupa pelebaran *range* karakter sesuai dengan karakter-karakter yang digunakan oleh algoritma *encoding* base64. *Encoding* base64 pada aplikasi ini digunakan untuk mengubah citra digital yang merupakan data *binary* menjadi berbentuk karakter-karakter tertentu sesuai dengan format karakter yang disediakan oleh *encoding* base64.

2.1 Vigenere Cipher

Vigenere Cipher merupakan hasil pengembangan lebih lanjut dari Caesar Cipher dan termasuk dalam kategori *polyalphabetic substitution cipher* (Prabowo & Hangga, 2015). Vigenere Cipher dapat dilakukan dengan dua cara yaitu dengan cara manual memakai bujur sangkar vigenere (*tabula recta*) seperti pada Gambar 1 maupun dengan cara substitusi angka (matematis). Secara matematis, enkripsi dan dekripsi menggunakan Vigenere Cipher dalam kondisi standar dapat dituliskan seperti Pers. (1), sedangkan untuk dekripsi bisa dituliskan seperti Pers. (2).

$$C_i = (P_i + K_i)\ mod\ 26 \qquad (1)$$

$$P_i = (C_i - K_i)\ mod\ 26 \qquad (2)$$

Berikut ini kami sajikan contoh penggunaan Vigenere Cipher dengan berpedoman pada susunan alfabet seperti pada Gambar 2.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Gambar 2 Indeks Susunan Huruf Alfabet

Plaintext : UINSUNANKALIJAGA
Key : VIGENERECIPHERVI
Ciphertext : PQTWHRRMIAPNRBI

Penelitian ini menggunakan susunan huruf yang digunakan pada *encoding* base64 sebanyak 64 karakter ditambah dengan tiga karakter tambahan yaitu ":", ";", dan "," karena berkas citra digital yang sudah di-*encode* menggunakan metode base64 memerlukan informasi tambahan berupa susunan *string mime type* sebagai penanda jenis berkas yang di-*encode*. Oleh karena itu, rumus matematis yang digunakan untuk proses enkripsi dan dekripsi juga berubah. Rumus persamaan untuk proses enkripsi sebagaimana tertulis pada Pers. (3). Sedangkan untuk proses dekripsi persamaan matematisnya sebagaimana tertulis pada Pers. (4).

$$C_i = (P_i + K_i)\ mod\ 67 \qquad (3)$$

$$P_i = (C_i - K_i)\ mod\ 67 \qquad (4)$$

2.2 Base64

Base64 merupakan salah satu skema *encoding binary-to-text* yang merepresentasikan data *binary* menjadi susunan karakter berformat ASCII, dengan menerjemahkannya ke dalam susunan radix-64 (Wen & Dang, 2018). Base64 merupakan suatu algoritma *block cipher* yang beroperasi dalam lingkup bit, hanya saja lebih mudah diimplementasikan dibandingkan algoritma yang lain. Karakter-karakter dalam radix-64 ini terdiri dari huruf "A-Z", "a-z", "0-9", dan dua karakter terakhir yaitu "/" dan "+" (Sumartono et al., 2016). Daftar karakter dalam radix-64 dapat dilihat pada Tabel 1.

Tabel 1 Susunan Karakter untuk Encoding Base64

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	/
15	P	31	f	47	v	63	+

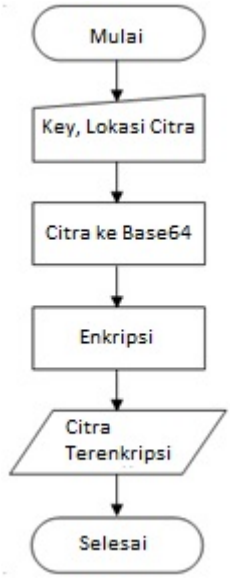
Berikut ini adalah contoh potongan *string* citra digital yang di-*encode* menggunakan base64 lengkap beserta dengan *mime type*:

GXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAMntJREFUeNrsfXtsHeeV35
Z …

2.3 Proses Enkripsi

Enkripsi merupakan proses untuk mengubah data asli menjadi data tersandi (Munir, 2006). Proses enkripsi tersaji dalam bentuk *flowchart* seperti yang terlihat pada Gambar 3.

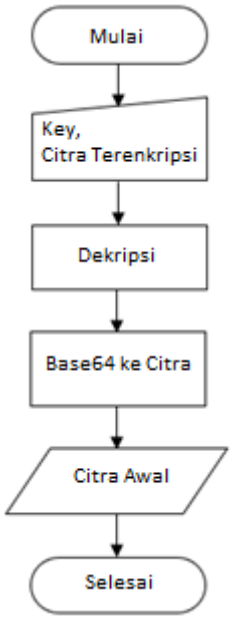
Langkah awal untuk melakukan proses enkripsi adalah dengan cara memasukkan *key*/*password* dan *path* lokasi *file* citra digital yang akan dienkripsi. Selanjutnya aplikasi akan melakukan konversi citra digital menjadi *encoded string* dengan metode base64. Kemudian aplikasi akan melakukan proses enkripsi dengan menerapkan Vigenere Cipher yang menggunakan susunan tabel Vigenere berdasarkan radix-64 ditambah tiga karakter lain dan *key* yang sudah dimasukkan. Sampai tahap tersebut, proses enkripsi selesai dilakukan dan hasilnya dikeluarkan menjadi berkas yang memiliki ekstensi *.vig.



Gambar 3 Diagram Alir Proses Enkripsi

2.4 Proses Dekripsi

Dekripsi merupakan proses untuk mengembalikan data yang sudah dienkripsi menjadi data asli (Munir, 2006). Proses dekripsi tersaji dalam bentuk *flowchart* seperti terlihat pada Gambar 4.



Gambar 4 Diagram Alir Proses Dekripsi

Langkah awal untuk melakukan dekripsi adalah pengguna memasukkan *key* dan *path* lokasi *file* yang akan didekripsi. Kemudian aplikasi akan melakukan proses dekripsi sebagaimana proses enkripsinya, yaitu menerapkan Vigenere Cipher menggunakan susunan tabel Vigenere berdasarkan radix-64 dengan tiga karakter tambahan dan *key* yang sudah dimasukkan. Selanjutnya aplikasi akan mengembalikan berkas citra digital yang dienkripsi menjadi *encoded string* berbasis base64. Kumpulan *string* tersebut akan di-*decode* dan dikeluarkan menjadi berkas citra digital sesuai dengan format aslinya berdasarkan *mime type*-nya.

3. HASIL DAN PEMBAHASAN

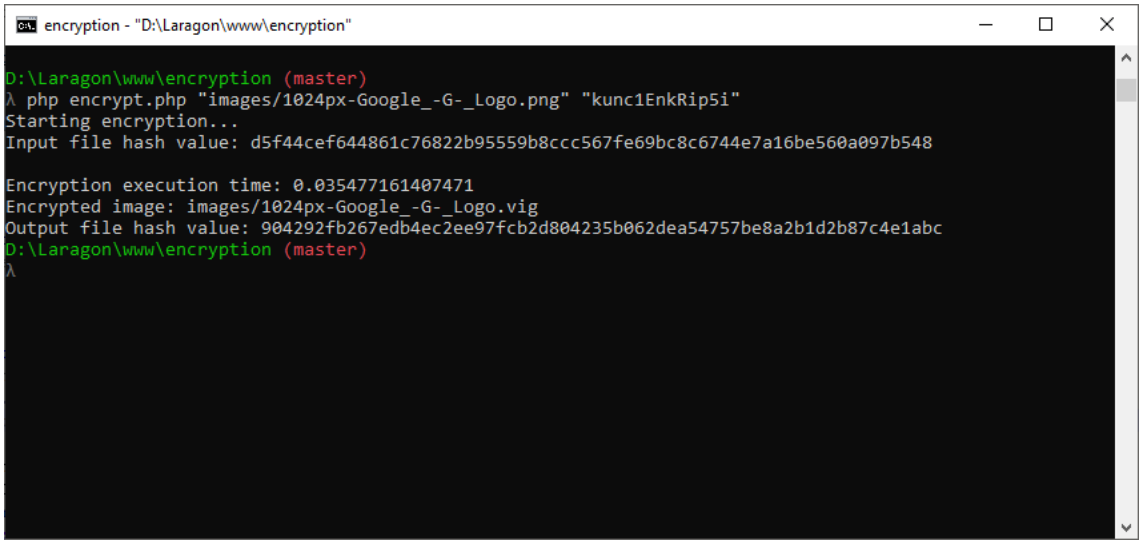
3.1 Modifikasi Vigenere Cipher

Penelitian ini menggunakan Vigenere Cipher yang sudah dimodifikasi. Modifikasi yang dilakukan terbatas pada pelebaran *support* karakter dari yang semula hanya 26 karakter alfabet menjadi 64 karakter yang terkandung dalam *list* radix-64 ditambah dengan karakter “:”, “;”, “,” sehingga berjumlah 67 karakter. Berikut adalah susunan karakter yang digunakan pada modifikasi Vigenere Cipher ini:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
+/:;,

3.2 Implementasi

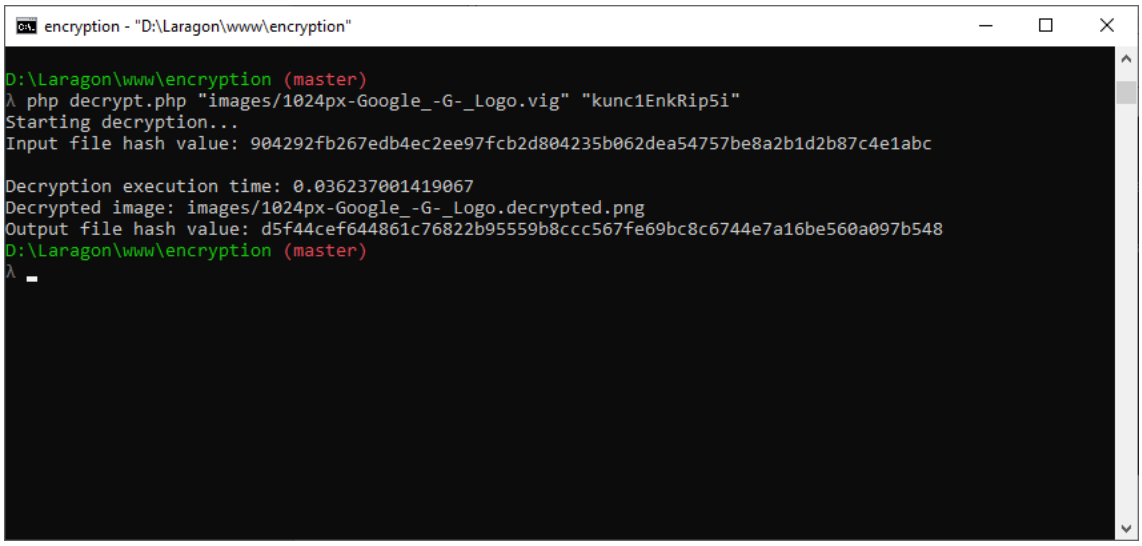
Penelitian ini menghasilkan *output* berupa aplikasi berbasis konsol dengan kata lain dijalankan melalui *terminal*/*command prompt* dan dibangun menggunakan bahasa pemrograman PHP versi 7.2.19 di atas platform Windows 10 Pro 64-bit. Aplikasi ini terdiri dari dua *file* utama yaitu encrypt.php dan decrypt.php, penamaan file ini sesuai dengan fungsi utama yang diusungnya. Sedangkan *core* utama untuk proses enkripsi-dekripsi menggunakan satu *file* saja yaitu VigenereCipher.php yang berada di dalam *folder source*. Contoh tampilan dalam proses enkripsi dan format *command* yang dijalankan seperti yang terlihat pada Gambar 5.



Gambar 5 Tampilan Proses Enkripsi

Terdapat dua *arguments* atau parameter yang wajib diisi saat menjalankan aplikasi ini, baik dalam proses enkripsi maupun dekripsi. Parameter pertama yaitu *string* yang berisi lokasi *file* baik yang akan dienkripsi maupun yang akan didekripsi. Parameter kedua yaitu *string* yang berisi *key* yang

digunakan. Contoh tampilan dalam proses dekripsi mirip dengan tampilan pada proses enkripsi seperti yang terlihat pada Gambar 6.



Gambar 6 Tampilan Proses Dekripsi

Aplikasi yang dihasilkan secara khusus hanya menerima input *file* citra digital dengan format png. Hasil dari proses enkripsi dikeluarkan ke dalam *file* dengan format ekstensi .vig, sedangkan hasil dari proses dekripsi dikeluarkan ke dalam *file* dengan format ekstensi .decrypted.png. Selain menghasilkan *output* berupa *file*, aplikasi ini juga menampilkan *output* nilai *hash* dari *file* yang akan dienkripsi, *file* yang sudah dienkripsi, dan *file* hasil dekripsi. Nilai *hash* ini digunakan untuk pengujian akurasi proses enkripsi dan dekripsi.

3.3 Pengujian

Pengujian dilakukan dengan menggunakan notebook Lenovo seri V14-ARE dengan spesifikasi *hardware* seperti yang ditunjukkan pada Tabel 2.

Tabel 2 Spesifikasi Hardware untuk Pengujian

Perangkat	Spesifikasi
Processor	AMD Ryzen 7 4700U with Radeon Graphics (8 CPUs), ~2.0GHz
Memory/RAM	1228 MB RAM
Display	AMD Radeon(TM) Graphics
Harddisk	1 TB
SSD	512 MB

Validitas hasil penelitian diuji dengan cara membandingkan nilai *hash file* asli sebelum dienkripsi dengan nilai *hash file* hasil dekripsi menggunakan SHA256. Pengujian dilakukan pada sepuluh *file* ctra digital yang berbeda dengan hasil seperti yang terlihat pada Tabel 3. *File* dengan format ekstensi .png atau .jpg merupakan *file* asli, sedangkan *file* dengan format ekstensi .vig merupakan *file* hasil enkripsi. *File* dengan format ekstensi .dec.png atau .dec.jpg merupakan *file* hasil dekripsi.

Perbandingan nilai *hash file* asli dengan nilai *hash file* hasil dekripsi selalu menunjukan tingkat kemiripan sebesar 100%. Hal ini berarti bahwa proses enkripsi dan dekripsi berhasil dengan baik dan akurat dalam mengamankan *file* citra digital.

Tabel 3 Pengujian Tingkat Akurasi

No	Nama *File*	Nilai *Hash*
1.	1024px.png	d5f44cef644861c76822b95559b8ccc567fe69bc8c6744e7a16be560a097b548
	1024px.vig	904292fb267edb4ec2ee97fcb2d804235b062dea54757be8a2b1d2b87c4e1abc
	1024px.dec.png	d5f44cef644861c76822b95559b8ccc567fe69bc8c6744e7a16be560a097b548
	Kemiripan	100%
2.	768px.png	9bb8b08d1d5dbbbc0cf1b81d5cad5e94bd05631370971a9fe8334ed97b0d8cfc
	768px.vig	a5f57fc69fef9b04bc36175a7354163017fbe34019d933903ddfb8d703dd8945
	768px.dec.png	9bb8b08d1d5dbbbc0cf1b81d5cad5e94bd05631370971a9fe8334ed97b0d8cfc
	Kemiripan	100%
3.	600px.png	6c4bb7f7ebf1e5531eb622421750eddc87d5c240f61e5960a70bb3bb9ac4ce43
	600px.vig	5efd0374b39c3d20fa4c38323bf2c24d018d68fcc5080145b675db6ffd79ef6f
	600px.dec.png	6c4bb7f7ebf1e5531eb622421750eddc87d5c240f61e5960a70bb3bb9ac4ce43
	Kemiripan	100%
4.	480px.png	77c2b739c04e67a345016430efd6831a084f65bcd300681cc0014c405b4c3a3d
	480px.vig	2e978fba910bc613d62f8f222d7cd14ebbd9dec0a0bdc3385631ab90dc980e7f
	480px.dec.png	77c2b739c04e67a345016430efd6831a084f65bcd300681cc0014c405b4c3a3d
	Kemiripan	100%
5.	240px.png	463be7410c34f9e9b7e078f832a73d91c69de520c6b3e162394658c8838be0b2
	240px.vig	df0ce75f56bba5ee7a90c1a51244217c6c541d8800dbbcc00c49276810ab3487
	240px.dec.png	463be7410c34f9e9b7e078f832a73d91c69de520c6b3e162394658c8838be0b2
	Kemiripan	100%
6.	jpeg1.jpg	67c0e8280d2d48a291d289784a6eb6d60782d87092200231079f119cbaf1169a
	jpeg1.vig	a06f1d7663447395f4a5b4c62872183cec30315338d2296f5a558b5f0faee6ec
	jpeg1.dec.jpg	67c0e8280d2d48a291d289784a6eb6d60782d87092200231079f119cbaf1169a
	Kemiripan	100%
7.	jpeg2.jpg	63fdb834976b4e7e74b0b1dbc18c9c5fa254916c922cdc434a30c611e41619cb
	jpeg2.vig	b23068d3a94950a1c3cf0a1beea18acfde2073f5188a112ea1237b685609594f
	jpeg2.dec.jpg	63fdb834976b4e7e74b0b1dbc18c9c5fa254916c922cdc434a30c611e41619cb
	Kemiripan	100%
8.	jpeg3.jpg	cc6d678810c48d8521864a456f28785170c52d76ff144dc7e34c7b2cfc2f090a
	jpeg3.vig	5363c0df7f3cb611961d6b97bd5b2b466bea0eb727d701f0505dc6bf699e0263
	jpeg3.dec.jpg	cc6d678810c48d8521864a456f28785170c52d76ff144dc7e34c7b2cfc2f090a
	Kemiripan	100%
9.	jpeg4.jpg	7ecd178674861ef3074a820f94bed5fb09894ff0115dfb704516afce1283233f
	jpeg4.vig	de5cc10114a879b01d098d56af3c26c510d1285f7d0d4d00c978c81d37c81fc3
	jpeg4.dec.jpg	7ecd178674861ef3074a820f94bed5fb09894ff0115dfb704516afce1283233f
	Kemiripan	100%
10.	jpeg5.jpg	5fd792469110989c8f332270726773cc90f649b22ea4649e4e6f6e0298d0d3fc
	jpeg5.vig	83b48381294265614017aa30dacc737a740437981827fdef39bd45fb8ff0fa62
	jpeg5.dec.jpg	5fd792469110989c8f332270726773cc90f649b22ea4649e4e6f6e0298d0d3fc
	Kemiripan	100%

Tabel 4 menyajikan hasil pengujian lain yang dilakukan yaitu penghitungan persentase rata-rata perubahan ukuran *file* yang dihasilkan dari membandingkan ukuran *file* asli dengan ukuran *file* setelah dienkripsi. Pengujian yang dilakukan pada sepuluh *file* citra digital yang berbeda-beda baik dari segi ukuran *file* maupun dimensi menunjukkan bahwa terjadi penambahan ukuran *file* rata-rata sebesar 33,34%. Penambahan ukuran rata-rata sebesar 33,34% ini tentu tidaklah buruk jika dibandingkan penggunaan algoritma Elgamal dengan mode operasi Electronic Code Book (ECB) untuk melakukan enkripsi citra digital seperti yang dilakukan oleh (Rizal, dkk., 2016).

Pengujian lain dilakukan dengan mengukur lama proses enkripsi dan dekripsi. Hasil uji lama proses enkripsi dan dekripsi tersaji pada Tabel 5. Pengujian yang dilakukan pada sepuluh *file* yang berbeda-beda baik dari segi ukuran *file* maupun dimensi menunjukkan bahwa proses enkripsi dilakukan dengan cepat dan membutuhkan waktu kurang dari 0,2 detik. Begitu pula dengan waktu yang digunakan untuk proses dekripsi yang tak terpaut jauh dengan proses

enkripsi yaitu kurang dari 0,19 detik. Hal ini tentu saja merupakan hasil yang bagus jika dibandingkan dengan penggunaan algoritma Arnold's Cat Map oleh (Rachmawanto, dkk., 2019).

Tabel 4 Pengujian Perubahan Ukuran File Hasil Enkripsi

No	Nama *File*	Dimensi	Ukuran *File*	Ukuran *Cipher*
1.	1024px.png	1024 x 1024	41,1 KB	54,9 KB
2.	768px.png	768 x 768	39,8 KB	53,1 KB
3.	600px.png	600 x 600	29,5 KB	39,3 KB
4.	480px.png	480 x 480	22,6 KB	30,1 KB
5.	240px.png	240 x 240	10,0 KB	13,4 KB
6.	jpeg1.jpg	1280 x 853	239 KB	318 KB
7.	jpeg2.jpg	1280 x 870	180 KB	240 KB
8.	jpeg3.jpg	1920 x 1020	89 KB	118 KB
9.	jpeg4.jpg	1280 x 854	309 KB	413 KB
10.	jpeg5.jpg	2480 x 1388	401 KB	535 KB

Tabel 5 Pengujian Lama Proses Enkripsi dan Dekripsi

No	Nama *File*	Dimensi	Lama Enkripsi	Lama Dekripsi
1.	1024px.png	1024 x 1024	0,036212921142578	0,041611194610596
2.	768px.png	768 x 768	0,036642074584961	0,038815975189209
3.	600px.png	600 x 600	0,028040885925293	0,026721954345703
4.	480px.png	480 x 480	0,020203828811646	0,020391941070557
5.	240px.png	240 x 240	0,010972023010254	0,0097010135650635
6.	jpeg1.jpg	1280 x 853	0.12175107002258	0.11409497261047
7.	jpeg2.jpg	1280 x 870	0.0935959815979	0.088201999664307
8.	jpeg3.jpg	1920 x 1080	0.046401023864746	0.043292999267578
9.	jpeg4.jpg	1280 x 854	0.15549278259277	0.1446430683136
10.	jpeg5.jpg	2480 x 1388	0.19454002380371	0.18740081787109

4. KESIMPULAN

Beberapa kesimpulan yang bisa didapatkan dari penelitian ini antara lain:

a. Implementasi algoritma Vigenere Cipher untuk proses enkripsi *file* citra digital berhasil dilakukan. Hal ini dapat dibuktikan dari hasil uji validitas yang selalu memperoleh skor kemiripan sebesar 100% antara *file* citra digital asli dengan *file* citra digital hasil dekripsi.
b. Proses enkripsi dan dekripsi *file* citra digital menggunakan algoritma Vigenere Cipher ini dilakukan dengan sangat cepat, terlihat dari proses enkripsi dan dekripsi yang memakan waktu tidak sampai dengan satu detik pada sepuluh sampel *file* uji.
c. *File* hasil enkripsi yang dihasilkan memiliki ukuran yang lebih besar rata-rata sebesar 33,34% dibanding dengan ukuran *file* aslinya.

Penggunaan *encoding* base64 sangat membantu menjembatani kekurangan algoritma Vigenere Cipher yang semula hanya mampu menangani data berupa teks. Penggunaan *encoding* ini dapat membantu proses enkripsi-dekripsi menggunakan algoritma Vigenere Cipher untuk *file* citra digital. Penelitian selanjutnya dapat dilakukan mengimplementasikan algoritma Vigenere Cipher ini untuk *file* dengan format yang beragam. Ide lain yang bisa dilakukan yaitu membandingkan kecepatan proses enkripsi maupun dekripsi dari algoritma Vigenere Cipher dengan algoritma kriptografi yang lain, termasuk dengan algoritma kriptografi modern.

DAFTAR PUSTAKA

Anwar, F., Rachmawanto, E. H., Atika Sari, C., & Ignatius Moses Setiadi, D. R. (2019). StegoCrypt Scheme using LSB-AES Base64. *2019 International Conference on Information and Communications Technology (ICOIACT)*, *July*, 85–90. https://doi.org/10.1109/ICOIACT46704.2019.8938567

Ariyus, D. (2006). *Kriptografi: Keamanan Data dan Komunikasi*. Graha Ilmu.

Awad, M., Ali, M., Takruri, M., & Ismail, S. (2019). Security vulnerabilities related to web-based data. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *17*(2), 852. https://doi.org/10.12928/telkomnika.v17i2.10484

Fadlil, A., Riadi, I., & Nugrahantoro, A. (2020a). Kombinasi Sinkronisasi Jaringan Syaraf Tiruan dan Vigenere Cipher untuk Optimasi Keamanan Informasi. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, *11*(1), 81–95. https://doi.org/10.31849/digitalzone.v11i1.3945

Fadlil, A., Riadi, I., & Nugrahantoro, A. (2020b). Data Security for School Service Top-Up Transactions Based on AES Combination Blockchain Technology. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, *11*(3), 155. https://doi.org/10.24843/LKJITI.2020.v11.i03.p04

Gerhana, Y. A., Insanudin, E., Syarifudin, U., & Zulmi, M. R. (2016). Design of digital image application using vigenere cipher algorithm. *2016 4th International Conference on Cyber and IT Service Management*, 1–5. https://doi.org/10.1109/CITSM.2016.7577571

Gunadhi, E., & Sudrajat, A. (2017). Pengamanan Data Rekam Medis Pasien Menggunakan Kriptografi Vigènere Cipher. *Jurnal Algoritma*, *13*(2), 295–301. https://doi.org/10.33364/algoritma/v.13-2.295

Hermansa, Umar, R., & Yudhana, A. (2019). Analisis Sistem Keamanan Teknik Kriptografi dan Steganografi Pada Citra Digital (Bitmap). *Seminar Nasional Teknologi Fakultas Teknik Universitas Krisnadwipayana*, 520–528.

Hernawandra, P., Supriyadi, S., & Lenggana, U. T. (2018). Aplikasi Steganografi Menggunakan LSB 4 Bit Sisipan dengan Kombinasi Algoritme Substitusi dan Vigenere Berbasis Android. *Jurnal Teknologi Dan Sistem Komputer*, *6*(2), 44–50. https://doi.org/10.14710/jtsiskom.6.2.2018.44-50

Mandal, S. K., & Deepti, A. R. (2016). A Cryptosystem Based On Vigenere Cipher By Using Mulitlevel Encryption Scheme. *International Journal of Computer Science and Information Technologies*, *7*(4), 2096–2099.

Maruf, F., Riadi, I., & Prayudi, Y. (2015). Merging of Vigenére Cipher with XTEA Block Cipher to Encryption Digital Documents. *International Journal of Computer Applications*, *132*(1), 27–33. https://doi.org/10.5120/ijca2015907262

Munir, R. (2006). *Kriptografi*. Informatika.

Nasution, S. D., Ginting, G. L., Syahrizal, M., & Rahim, R. (2017). Data Security Using Vigenere Cipher and Goldbach Codes Algorithm. *International Journal of Engineering Research & Technology (IJERT)*, *6*(1), 360–363.

Prabowo, H. E., & Hangga, A. (2015). Enkripsi Data Berupa Teks Menggunakan Metode Modifikasi Vigenere Cipher. *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, 1–4.

Rachmawanto, E. H., Irawan, C., Studi, P., Informatika, T., Komputer, F. I., Dian, U., Semarang, N., & Digital, C. (2019). Enkripsi Dan Dekripsi Citra RGB Menggunakan Algoritma Arnold's Cat Map. *Prosceeding SENDI_U*, 44–49.

Riadi, I., Yudhana, A., & W, Y. (2020). Analisis Keamanan Website Open Journal System Menggunakan Metode Vulnerability Assessment. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *7*(4), 853. https://doi.org/10.25126/jtiik.2020701928

Rihartanto, R., Ningsih, R. K., Gaffar, A. F. O., & Utomo, D. S. B. (2020). Implementation of vigenere cipher 128 and square rotation in securing text messages. *Jurnal Teknologi Dan Sistem Komputer*, *8*(3), 201–209. https://doi.org/10.14710/jtsiskom.2020.13476

Rizal, D., Sutojo, T., & Rahayu, Y. (2016). Implementasi Kriptografi Gambar Menggunakan Kombinasi Algoritma Elgamal Dan Mode Operasi ECB (Electronic Code Book). *Techno.COM*, *15*(3), 231–240. https://doi.org/10.33633/tc.v15i3.1240

Rojali, Salman, A. G., & George. (2017). Website-based PNG image steganography using the modified Vigenere Cipher, least significant bit, and dictionary based compression methods. *International Conference on Mathematics: Pure, Applied and Computation*, 020059. https://doi.org/10.1063/1.4994462

Saputra, I., Hasibuan, N. A., Aan, M., & Rahim, R. (2017). Vigenere Cipher Algorithm with Grayscale Image Key Generator for Secure Text File. *International Journal of Engineering Research & Technology (IJERT)*, *6*(1), 266–269.

Setiadi, D. R. I. M., Jatmoko, C., Rachmawanto, E. H., & Sari, C. A. (2018). Kombinasi Cipher Subtitusi (Beaufort Dan Vigenere) pada Citra Digital. *Prosceeding SENDI_U*, 52–57.

Sinaga, D., Umam, C., Setiadi, D. R. I. M., & Rachmawanto, E. H. (2018). Teknik Super Enkripsi Menggunakan Transposisi Kolom Berbasis Vigenere Cipher Pada Citra Digital. *Dinamika Rekayasa*, *14*(1), 57. https://doi.org/10.20884/1.dr.2018.14.1.198

Soofi, A. A., Riaz, I., & Rasheed, U. (2016). An Enhanced Vigenere Cipher For Data Security. *International Journal of Scientific & Technology Research*, *5*(03), 141–145.

Subandi, A., Lydia, M. S., Sembiring, R. W., Zarlis, M., & Efendi, S. (2018). Vigenere cipher algorithm modification by adopting RC6 key expansion and double encryption process. *IOP Conference Series: Materials Science and Engineering*, *420*, 012119. https://doi.org/10.1088/1757-899X/420/1/012119

Sumartono, I., Siahaan, A. P. U., & Arpan. (2016). Base64 Character Encoding and Decoding Modeling. *International Journal of Recent Trends in Engineering & Research (IJRTER)*, *02*(12), 63–68. https://doi.org/10.31227/osf.io/ndzqp

Wen, S., & Dang, W. (2018). Research on Base64 Encoding Algorithm and PHP Implementation. *2018 26th International Conference on Geoinformatics*, 1–5. https://doi.org/10.1109/GEOINFORMATICS.2018.8557068

Yunita, S., Hasan, P., & Ariyus, D. (2019). Modifikasi Algoritma Hill Chiper dan Twofish Menggunakan Kode Wilayah Telepon. *SISFOTENIKA*, *9*(2), 213. https://doi.org/10.30700/jst.v9i2.489

Zebua, T., & Ndruru, E. (2017). Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *4*(4), 275–282. https://doi.org/10.25126/jtiik.201744474

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 1, JANUARI, 2022, Pp. 46 – 55
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Pemanfaatan *Network Forensic Investigation Framework* untuk Mengidentifikasi Serangan Jaringan Melalui *Intrusion Detection System* (IDS)

Tri Widodo [(1)*], Adam Sekti Aji [(2)]
[1] Pendidikan Teknologi Informasi, Fakultas Bisnis dan Humaniora, Universitas Teknologi Yogyakarta, Yogyakarta
[2] Informatika, Fakultas Sains dan Teknologi, Universitas Teknologi Yogyakarta, Yogyakarta
e-mail : triwidodo@uty.ac.id, adamaji@staff.uty.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 8 September 2021, direvisi 15 November 2021, diterima 15 November 2021, dan dipublikasikan 25 Januari 2022.

Abstract

Intrusion Detection System (IDS) is one of the technology to ensure the security of computers. IDS is an early detection system in the event of a computer network attack. The IDS will alert the computer network administrator in the event of a computer network attack. IDS also records all attempts and activities aimed at disrupting computer networks and other computer network attacks. The purpose of this study is to implement IDS on network systems and analyze IDS logs to determine the different types of computer network attacks. Logs on the IDS will be analyzed and will be used as leverage to improve computer network security. The research was carried out using the Network Forensic Investigation Framework proposed by Pilli, Joshi, and Niyogi. The stages of the Network Forensic Investigation Framework are used to perform network simulations, analysis, and investigations to determine the types of computer network attacks. The results show that the Network Forensic Investigation Framework facilitates the investigation process when a network attack occurs. The Network Forensic Investigation Framework is effectively used when the computer network has network security support applications such as IDS or others. IDS is effective in detecting network scanning activities and DOS attacks. IDS gives alerts to administrators because there are activities that violate the rules on the IDS.

Keywords: Network Forensic Investigation Framework, Intrusion Detection System (IDS), Network Attack, Network Scanning, DOS Attacks

Abstrak

Salah satu media untuk mengamankan komputer adalah menerapkan teknologi *Intrusion Detection System* (IDS). IDS merupakan sistem deteksi dini jika terjadi serangan jaringan komputer. IDS akan memberi peringatan kepada administrator jaringan komputer jika terjadi serangan jaringan komputer. IDS juga mencatat semua upaya dan kegiatan-kegiatan yang bertujuan mengganggu jaringan komputer maupun serangan jaringan komputer lainnya. Tujuan penelitian ini yaitu mengimplementasikan IDS pada sistem jaringan dan menganalisis catatan (*log*) IDS untuk mengetahui jenis-jenis dan tipe serangan jaringan komputer. *Log* pada IDS akan dianalisis secara mendalam untuk digunakan sebagai upaya meningkatkan keamanan jaringan komputer. Metode penelitian yang akan digunakan adalah penelitian terapan (*applied research*). Pelaksanaan penelitian menggunakan *Network Forensic Investigation Framework* yang dikemukakan oleh Pilli, Joshi dan Niyogi. Tahapan-tahapan *Network Forensic Investigation Framework* digunakan untuk melakukan simulasi jaringan, analisis dan investigasi untuk mengetahui jenis-jenis serangan jaringan komputer. Hasil penelitian menunjukkan *Network Forensic Investigation Framework* memudahkan proses investigasi ketika terjadi serangan jaringan. *Network Forensic Investigation Framework* efektif digunakan ketika jaringan komputer memiliki aplikasi pendukung keamanan jaringan seperti IDS atau yang lainnya. IDS efektif mendeteksi adanya aktivitas *network scanning* dan serangan DOS. IDS memberikan alert pada administrator karena ada aktivitas yang melanggar *rule* pada IDS.

Kata Kunci: *Network Forensic Investigation Framework*, *Intrusion Detection System* (IDS), *Network Attack*, *Network Scanning*, Serangan DOS

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 1, JANUARI, 2022: 46 – 55

1. PENDAHULUAN

Internet merupakan kebutuhan yang sangat penting pada era digital seperti sekarang. Revolusi industri 4.0 mengharuskan setiap orang terhubung ke jaringan internet setiap saat untuk berkomunikasi. Institusi atau perusahaan menjadikan internet sebagai bagian dari infrastruktur untuk meningkatkan produktivitas karyawan dan perusahaan. Tingginya kebutuhan internet terkadang dimanfaatkan oleh pihak-pihak tertentu untuk serangan jaringan komputer. Serangan jaringan komputer meningkat secara signifikan pada era digital seperti ini. Pusat Operasi keamanan Siber Nasional (Pusopskamsinas) Badan Siber dan Sandi Negara (BSSN) mencatat ada 88.414.296 serangan siber di Indonesia yang terjadi sejak 1 Januari hingga 12 April 2020. Pada Januari terpantau ada 25.224.811 serangan dan kemudian pada Februari terekam 29.188.645 serangan. Lalu, pada Maret terjadi 26.423.989 serangan dan sampai dengan 12 April 2020 tercatat ada 7.576.851 serangan (Iskandar, 2020). Para *hacker* tidak hanya menargetkan serangan untuk melumpuhkan jaringan komputer suatu perusahaan. Mereka juga berusaha untuk mencuri berbagai data dari server.

Administrator jaringan komputer memiliki tugas dan tanggung jawab yang penting pada sebuah institusi atau perusahaan. Administrator jaringan bertanggung jawab atas desain, perencanaan, operasi, keamanan, dan manajemen sehari-hari dari jaringan, server, *switch*, jaringan internet organisasi dan semua komunikasi data. Salah satu tugas berat administrator jaringan komputer adalah mengamankan infrastruktur jaringan komputer dan data perusahaan dari serangan jaringan komputer. Serangan jaringan komputer atau *network attack* adalah upaya untuk mendapatkan akses tidak sah ke jaringan organisasi, dengan tujuan mencuri data atau melakukan aktivitas berbahaya lainnya. Intrusi adalah upaya tidak sah, upaya ilegal untuk mengakses, memanipulasi atau menguasai sistem/jaringan informasi untuk membuat mereka tidak dapat diandalkan atau tidak dapat digunakan (Kumar, 2017). Ada banyak bentuk implementasi keamanan jaringan mulai dari sistem AAA (*Authentication, Authorization & Accounting*), *Firewalls*, *Routing Filters*, *Access Control*, *Intrusion Prevention System*, *Intrusion Detection Systems*, *Honeypot*, dan lain-lain (Alsyaibani et al., 2021).

Administrator jaringan komputer dapat mengimplementasikan *Intrusion Detection System* (IDS) untuk mengetahui adanya serangan jaringan komputer (Lazzez, 2013). IDS akan memberikan *alert* (peringatan) kepada administrator jaringan jika terjadi serangan atau gangguan terhadap jaringan. Teknik pendeteksian pada IDS umum masih jauh dari sempurna jika dibandingkan dengan berbagai anomali dan alat modern yang digunakan oleh penyerang karena IDS masih menggunakan deteksi berbasis tanda tangan atau model deteksi berbasis anomali (Chowdhury et al., 2017).

Administrator jaringan dapat melakukan analisis terhadap catatan (*log*) yang direkam oleh IDS. Hasil analisis *log* IDS dapat digunakan untuk mengetahui jenis-jenis serangan jaringan komputer yang ditujukan ke jaringan komputer, sehingga administrator jaringan dapat melakukan perbaikan, pengaturan ulang jaringan, dan mengimplementasikan aplikasi-aplikasi tertentu untuk meningkatkan keamanan jaringan komputer yang dikelola.

Berbagai penelitian tentang *Intrusion Detection System* sudah pernah dilakukan. Penelitian yang dilakukan oleh Khaerani & Handoko (2015) dengan judul 'Implementasi dan Analisa *Data Mining* untuk Klasifikasi Serangan Pada *Intrusion Detection System* (IDS) dengan Algoritma C4.5' menggunakan data tahun 1999. Saat itu penggunaan internet belum semasif sekarang, jenis serangan jaringan komputer juga masih terbatas. Selanjutnya penelitian dilakukan Muhammad, (2016) yang secara khusus menganalisis *log* IDS berbasis *neural network* untuk mengetahui serangan DDOS. Sayangnya penelitian ini juga hanya mengkhususkan serangan jaringan DDOS, selain itu penelitian juga bersifat prediksi terhadap potensi serangan tersebut, sehingga penelitian sulit diimplementasikan dan tidak dapat mengetahui jenis-jenis serangan jaringan lainnya. Penelitian serupa juga dilakukan oleh Purba & Efendi (2021), penelitian ini juga berfokus pada serangan DDOS. Penelitian selanjutnya dilakukan oleh Suhartono & Patta (2017), para peneliti

melakukan penelitian dengan membatasi lingkup serangan pada serangan melalui dua *port* jaringan yaitu, SSH dan FTP sehingga membuat penelitian tersebut lebih menarik.

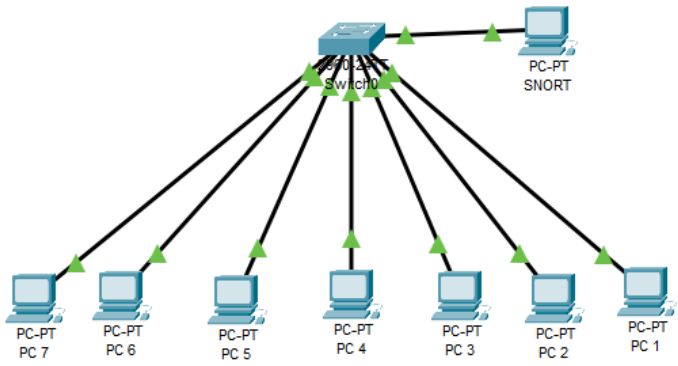
Penelitian yang dilakukan ini berfokus dalam mengimplementasikan konsep *Network Forensic Investigation Framework* dan Snort IDS dalam mendeteksi berbagai jenis serangan jaringan. Penelitian ini juga menggabungkan aspek keamanan jaringan komputer dengan aspek forensik digital. Penelitian ini tidak ditujukan untuk menguji IDS dengan serangan tertentu, tetapi difokuskan pada contoh pemanfaatan IDS dan juga bagaimana melakukan investigasi serangan jaringan secara efektif dan mudah menggunakan *Network Forensic Investigation Framework*.

2. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah penelitian terapan. Menurut Irina (2017) penelitian terapan atau *applied research* dilakukan berkenaan dengan kenyataan-kenyataan praktis penerapan dan pengembangan ilmu pengetahuan yang dihasilkan oleh penelitian dasar dalam kehidupan nyata. Penelitian terapan berfungsi untuk mencari solusi tentang masalah-masalah tertentu. Tujuan utamanya adalah pemecahan masalah sehingga hasil penelitian dapat dimanfaatkan untuk kepentingan manusia baik secara individu atau kelompok maupun untuk keperluan industri atau politik dan bukan untuk wawasan keilmuan semata. Penelitian ini akan menggunakan *Network Forensic Investigation Framework* yang dikemukakan Pilli et al. (2010) dengan menggunakan 9 (sembilan) tahapan.

2.1 *Preparation and Authorization* (Persiapan dan Otorisasi)

Pada tahap ini akan buat sistem jaringan yang terdiri dari 7 PC *client*, 1 buah *switch*, dan 1 PC yang ter-*install* Snort IDS seperti terlihat pada Gambar 1. *Framework* ini juga mengharuskan otorisasi dan akses penuh terhadap sistem jaringan dan Snort IDS.



Gambar 1 Desain Topologi Jaringan Simulasi

Spesifikasi perangkat pada topologi jaringan dapat dilihat pada Tabel 1.

Tabel 1 Spesifikasi Perangkat Jaringan

No	Perangkat	Alamat IP	Perangkat Lunak
1	Komputer Server	192.168.56.1	a. Sistem Operasi Ubuntu b. Snort IDS
2	Komputer *Client* 1	192.168.56.100	Windows 10
3	Komputer *Client* 2	192.168.56.101	Linux
4	Komputer *Client* 3	192.168.56.102	Windows 10
5	Komputer *Client* 4	192.168.56.103	Windows 10
6	Komputer *Client* 5	192.168.56.104	Windows 10
7	Komputer *Client* 6	192.168.56.105	Linux
8	Komputer *Client* 7	192.168.56.106	Windows 10
9	*Switch*	-	-

2.2 *Detection and Incident/Crime* (Deteksi Insiden/Kejahatan)

Setelah jaringan dan Snort IDS ter-*install* dan terkonfigurasi, akan dilakukan berbagai simulasi serangan jaringan. Uji coba akan menggunakan teknik serangan jaringan sederhana yaitu *network scanning* dan serangan DOS. IDS akan ditambahkan dengan *rule* untuk mendeteksi adanya *network scanning* dan serangan DOS. IDS akan menganalisis aktivitas jaringan berdasarkan *rule* yang diberikan dan memberikan *alert* (peringatan) kepada administrator jika ada aktivitas yang melanggar *rule*.

2.3 *Incident Response* (Penanganan Insiden)

Insiden respon dini yang dilakukan adalah adanya *alert* (peringatan) dari Snort IDS. Respon selanjutnya yang akan dilakukan adalah mengubah dan memodifikasi *rule* (aturan) pada Snort IDS. Perubahan beberapa *rule* digunakan untuk mengetahui kemampuan dan fungsi Snort IDS.

2.4 *Collection of Network Traces* (Koleksi Jejak Jaringan)

Setelah serangan jaringan komputer, dilakukan pelacakan dan verifikasi kesesuaian sumber serangan, jenis serangan, dan peringatan yang diberikan Snort IDS.

2.5 *Preservation and Protection* (Pengamanan dan Perlindungan Data)

Tahap ini berisi pengamanan data yang menjadi bukti serangan jaringan. Bukti ini berupa *log* (catatan) dan peringatan yang diberikan oleh Snort IDS. Pengamanan ini perlu dilakukan karena beberapa serangan dapat memungkinkan penyerang untuk menghapus jejak dan bukti.

2.6 *Examination* (Pemeriksaan)

Pemeriksaan data digunakan untuk memisahkan data aktivitas normal dan data serangan jaringan komputer. Pemeriksaan ini juga digunakan untuk mengetahui adanya sumber data lain selain *log* (catatan) IDS yang dapat digunakan untuk membantu proses analisis.

2.7 *Analysis* (Analisis)

Bukti-bukti yang sudah terkumpul, kemudian dianalisis menggunakan berbagai Teknik dan perangkat bantu (*tool*). Analisis ini menggunakan berbagai parameter seperti jenis koneksi jaringan, sistem operasi, dan protokol jaringan yang digunakan.

2.8 *Investigation and Attribution* (Investigasi dan Atribusi)

Hasil analisis kemudian diinvestigasi untuk mengetahui komputer mana yang melakukan serangan, siapa pemilik komputer tersebut, serangan jenis apa yang digunakan, dampak serangan tersebut bagi sistem, dan bagaimana cara mengatasi serangan tersebut selanjutnya.

2.9 *Presentation* (Presentasi)

Hasil analisis akan dibuat laporan dan penjelasan yang lebih mudah dipahami oleh berbagai pihak terkait. Presentasi juga memuat saran dan tindakan perbaikan yang berguna untuk meningkatkan tingkat keamanan jaringan.

3. HASIL DAN PEMBAHASAN

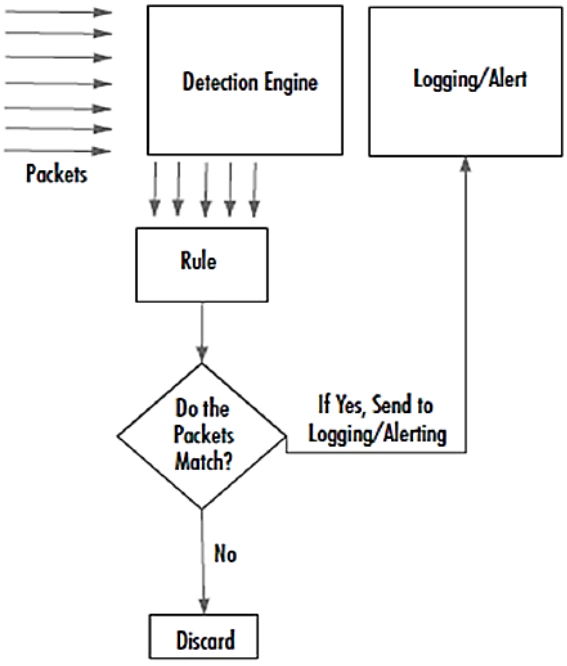
Intrusion Detection System (IDS) adalah perangkat atau aplikasi perangkat lunak yang memantau lalu lintas data pada jaringan komputer untuk mengetahui aktivitas berbahaya atau pelanggaran kebijakan (Barracuda Networks, 2021). Ada beberapa jenis IDS, yaitu:

1) *Network Intrusion Detection Systems* (NIDS), yaitu IDS yang menganalisis lalu lintas data jaringan komputer.
2) *Host-Based Intrusion Detection Systems* (HIDS), yaitu IDS yang memantau file sistem operasi.

Intrusion Detection System (IDS) dalam mendeteksi menggunakan metode *signature based* dan *anomaly based* (Alviana & Sumitra, 2018). Menurut Alviana & Sumitra (2018) metode *anomaly based* merupakan metode dalam mendeteksi serangan melalui pola lalu lintas jaringan di luar kebiasaan, sedangkan metode *signature based* merupakan metode dalam mendeteksi serangan melalui pola atau paket data yang dibaca kemudian dibandingkan dengan data atau paket yang sudah tersimpan dalam *database* yang ada atau *rule* yang sudah ada.

Snort merupakan salah satu sistem deteksi intrusi (IDS) *open source* yang banyak digunakan untuk mendeteksi intrusi atau aktivitas mencurigakan pada lalu lintas jaringan (Paramitha et al., 2020). Snort merupakan salah contoh program dari *Network-based Intrusion Detection System* (Sandi & Arrofiq, 2018). Cara kerja Snort mirip dengan TcpDump, tetapi fokus sebagai *security packet sniffing*. Fitur utama Snort yang membedakan dengan TcpDump adalah *payload inspection*, di mana Snort melakukan analisis *payload rule set* yang disediakan (Dewi, 2017).

Menurut Singh & Tomar (2015), Snort bekerja sebagai *detection engine* dengan IDS *mode*, Ketika terdapat *packet* datang melalui *switch* maka akan terdeteksi oleh *detection engine* pada Snort, kemudian Snort sebagai IDS akan mencocokkan *packet* tadi dengan dengan *rules* yang sudah diatur. Ketika *packet* tersebut tidak sesuai *rule* (*packet* mengandung konten serangan) maka akan tersimpan di *log* Snort dan akan membangkitkan *alarm*, namun jika *packet* tersebut sesuai *rule* (tidak mengandung konten serangan) maka *packet* tersebut di-*discard* (diabaikan) dan langsung diteruskan.



Gambar 2 Cara Kerja Mesin Deteksi Snort (Singh & Tomar, 2015)

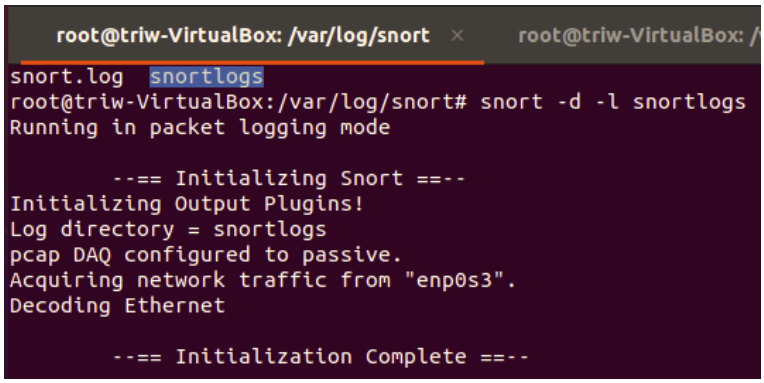
Data *log* IDS Snort ini dapat dimanfaatkan oleh administrator jaringan untuk menganalisis performa sistem keamanan jaringan (Paramitha et al., 2020).

Network Forensic Investigation Framework yang dikemukakan Pilli et al. (2010) yang menggunakan 9 (sembilan) tahapan, implementasinya dijelaskan sebagai berikut.

3.1 *Preparation and Authorization* (Persiapan dan Otorisasi)

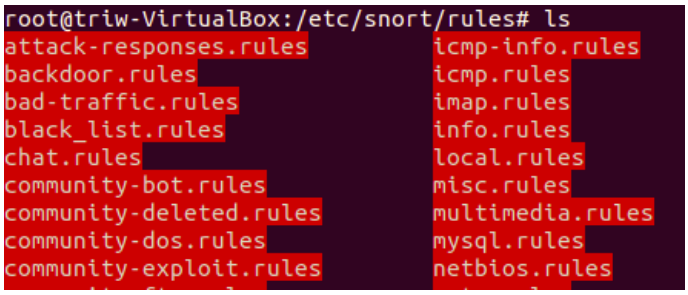
Jaringan komputer secara umum dilengkapi dengan berbagai aplikasi pengaman, seperti *firewall*, *anti-virus*, *proxy* atau IDS. Seorang administrator jaringan harus memiliki akses dan kendali

penuh terhadap jaringan komputer yang dikelola. Administrator juga memastikan berbagai aplikasi pengaman tersebut aktif. Pada Gambar 3 terlihat IDS Snort dalam posisi aktif dan merekam segala aktivitas jaringan komputer.



Gambar 3 Mengaktifkan *Logging Mode* pada Snort

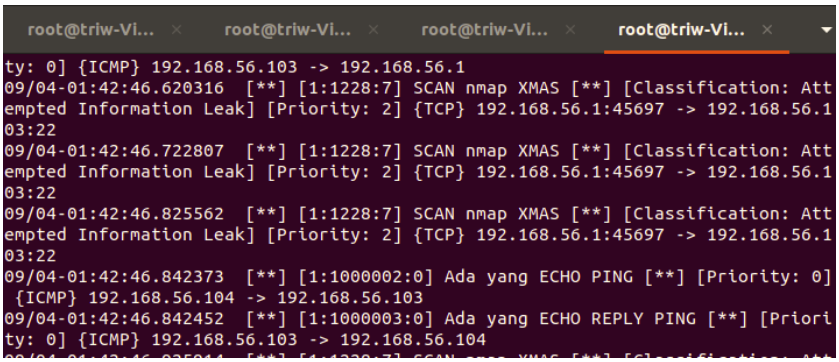
IDS Snort bekerja berdasarkan *rule* atau aturan yang ditentukan oleh administrator. *Rule* tersebut yang akan menjadi patokan kerja IDS, seperti menolak paket, meneruskan paket atau memberikan *alert*. *Rule* tersebut dapat dipisah menjadi beberapa aturan seperti terlihat pada Gambar 4.



Gambar 4 Konfigurasi *Rule* pada Snort

3.2 *Detection and Incident/Crime* (Deteksi Insiden/Kejahatan)

IDS Snort akan mendeteksi berbagai serangan berdasarkan pada *rule* yang telah ditentukan. Snort akan memberikan *alert* atau peringatan pada administrator terkait serangan atau akses tertentu pada jaringan seperti terlihat pada Gambar 5. Pada penelitian ini PC 4 mencoba melakukan enumerasi pada komputer server dengan melakukan *scanning*. Proses *scanning* tersebut dapat dengan baik dideteksi oleh Snort IDS. Selanjutnya PC 4 juga melakukan pengiriman *ping* secara terus menerus yang dapat diindikasikan sebagai serangan DOS.



Gambar 5 Alert yang Muncul pada *Console* yang Mendeteksi Adanya Serangan

3.3 *Incident Response* (Penanganan Insiden)

Ketika administrator menerima *alert* atau *alarm* dari IDS. Administrator dapat menindaklanjuti dengan beberapa Tindakan, seperti *blocking port* seperti pada Gambar 6, *blocking* IP, atau menonaktifkan beberapa *protocol*.

```
root@triw-VirtualBox:/home/triw# ufw deny 23/tcp
Rule updated
Rule updated (v6)
```

Gambar 6 Respon Berupa Penutupan *Port* Tertentu

Admisntrator juga dapat menindaklanjuti dengan merubah beberapa *rule* agar keamanan jaringan lebih optimal seperti Gambar 7.

```
root@triw-VirtualBo...   root@triw-VirtualBo...   root@triw-VirtualBo...
 GNU nano 4.8                        local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ---------------
# LOCAL RULES
# ---------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
#percobaan rule baru
log tcp any any -> 192.168.56.0/24 !6000:6010
alert tcp any any -> 192.168.56.103 23 (msg: "Ada yang telnet ke mesin!"; sid:>
alert icmp any any <> 192.168.56.103 any (msg:"Ada yang ECHO PING"; icode:0; i>
alert icmp any any <> 192.168.56.103 any (msg:"Ada yang ECHO REPLY PING"; icod>
```

Gambar 7 Penyesuaian dan Perubahan Rule Snort untuk Mengantisipasi Serangan Jaringan

3.4 *Collection of Network Traces* (Koleksi Jejak Jaringan)

Semua aktivitas jaringan akan direkam IDS Snort pada *log* atau catatan seperti terlihat pada Gambar 8.

```
root@triw-VirtualBox:/var/log/snort/snortlogs# ls -l
total 308184
-rw-r--r-- 1 root snort         0 Sep  4 01:21 alert
-rwsrwxr-t 1 root snort       787 Sep  1 17:16 snort.log.1630491357
-rw------- 1 root snort      1899 Sep  1 17:43 snort.log.1630492975
-rw------- 1 root snort  50641079 Sep  2 17:07 snort.log.1630566064
-rw------- 1 root snort 129963719 Sep  4 01:23 snort.log.1630689641
-rw------- 1 root snort 134216566 Sep  4 01:52 snort.log.1630694117
-rw------- 1 root snort    730999 Sep  4 02:55 snort.log.1630695145
```

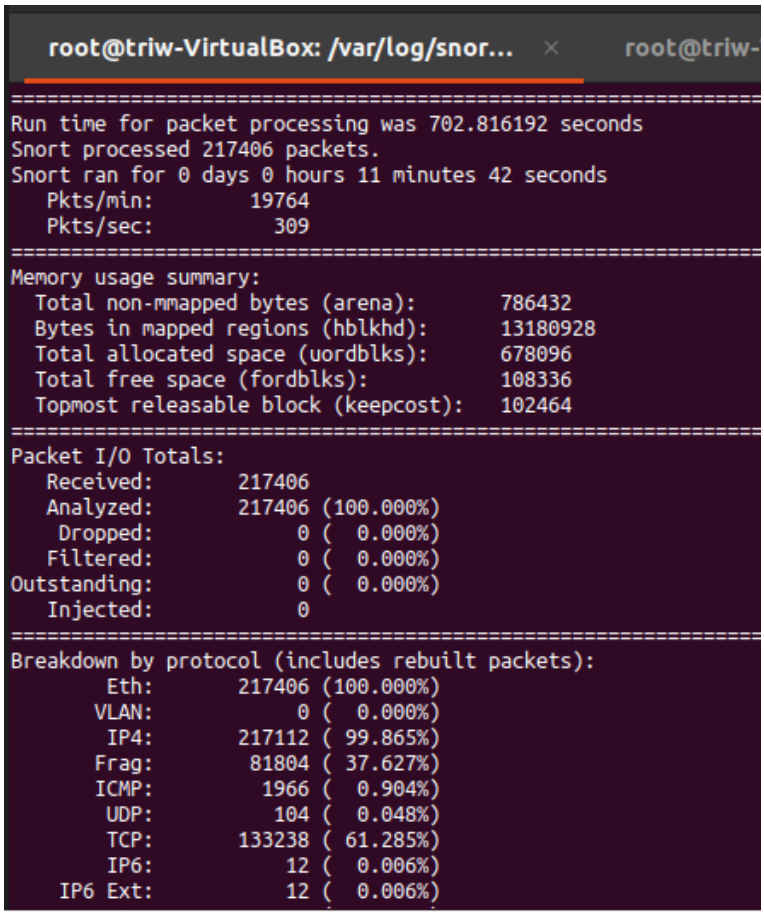
Gambar 8 *Log* (Catatan) Aktivitas *Log* yang Didapat dari Snort

3.5 *Preservation and Protection* (Pengamanan dan Perlindungan Data)

Tugas selanjutnya administrator ketika mengidentifikasi adanya serangan atau gangguan jaringan adalah mengamankan *log* atau catatan yang direkam oleh IDS Snort, karena catatan tersebut akan menjadi data penting bagi administrator untuk mengetahui jenis serangan, sumber serangan dan *protocol* yang menjadi sasaran.

3.6 *Examination* (Pemeriksaan)

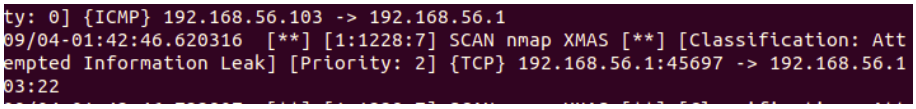
Administrator jaringan setelah mendapatkan catatan atau *log* IDS Snort selanjutnya melakukan pemeriksaan dan identifikasi aktivitas jaringan komputer. *Log* IDS Snort akan memberikan informasi aktivitas jaringan seperti besar paket maupun *protocol* yang digunakan seperti pada Gambar 9.



Gambar 9 Hasil Rekap Aktivitas Jaringan yang Dicatat Snort

3.7 *Analysis* (Analisis)

Administrator melalui *log* mengetahui sumber IP penyerang yaitu 192.168.56.103, aktivitas yang dilakukan adalah scanning port pada server 192.168.56.1 seperti pada Gambar 10.



Gambar 10 Identifikasi Sumber Serangan Jaringan

3.8 *Investigation and Attribution* (Investigasi dan Atribusi)

Administrator mengidentifikasi bahwa PC 4 192.168.56.103 telah melakukan pengiriman paket sebanyak 217406 kali dengan durasi 11 menit 42 detik. Administrator mengidentifikasi *host* dengan IP 192.168.56.103 juga telah mengirimkan paket sebesar 13180928 bytes atau sekkitar 206 MB yang dapat diindikasikan merupakan *packet flooding* atau jenis serangan DOS. PC 4 juga terindikasi melakukan enumerasi dengan melakukan *scanning.* Hal itu patut dicurigai bahwa PC 4 berusaha mencari celah keamanan dari komputer server.

3.9 *Presentation* (Presentasi)

Langkah *digital forensic* terakhir administrator adalah membuat laporan hasil investigasi agar dapat dilaporkan pada pimpinan dan ditindaklanjuti dengan pembaharuan keamanan jaringan.

4. KESIMPULAN

Penelitian ini menunjukkan *Network Forensic Investigation Framework* memudahkan proses investigasi ketika terjadi serangan jaringan. *Network Forensic Investigation Framework* efektif digunakan ketika jaringan komputer memiliki aplikasi pendukung keamanan jaringan seperti IDS atau yang lainnya. IDS efektif mendeteksi adanya aktivitas *network scanning* dan serangan DOS. IDS memberikan *alert* pada administrator karena ada aktivitas yang melanggar *rule* pada IDS. Catatan atau *log* IDS mempermudah proses investigasi sehingga serangan jaringan dapat terlacak sampai pada sumber serangan dan media serangan. Penelitian selanjutnya diharapkan dapat mengkolaborasikan perangkat keamanan jaringan dengan perangkat kecerdasan buatan atau *machine learning*. Penelitian-penelitian yang menggabungkan aplikasi keamanan jaringan komputer dan kecerdasan buatan atau *machine learning* sangat penting terutama untuk mendeteksi pornografi maupun serangan *malware*.

UCAPAN TERIMA KASIH

Terima kasih kami sampaikan kepada Direktorat Sumber Daya Direktorat Jenderal Pendidikan Tinggi yang telah memberikan dukungan penelitian melalui skema Penelitian Dosen Pemula tahun 2021 sesuai dengan Kontrak Penelitian Tahun Tunggal Penelitian Dasar dan Pembinaan/Kapasitas Tahun Anggaran 2021 dengan LLDIKTI Wilayah V Nomor 006/E4.1/AK.04.PT/2021, tanggal 12 Juli 2021 dan segenap pihak yang telah membantu penelitian ini.

DAFTAR PUSTAKA

Alsyaibani, O. M. A., Utami, E., & Hartanto, A. D. (2021). Survey on Deep Learning Based Intrusion Detection System. *Telematika*, *14*(2), 86–100. https://doi.org/10.35671/telematika.v14i2.1317

Alviana, S., & Sumitra, I. D. (2018). Analisis Pengukuran Penggunaan Sumber Daya Komputer pada Intrusion Detection System dalam Meminimalkan Serangan Jaringan. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, *7*(1), 27–34. https://doi.org/10.34010/komputa.v7i1.2533

Barracuda Networks. (2021). *What is an Intrusion Detection System?* Barracuda Networks, Inc. https://www.barracuda.com/glossary/intrusion-detection-system

Chowdhury, F. Z., Kiah, L. B. M., Ahsan, M. A. M., & Bin Idris, M. Y. I. (2017). Economic denial of sustainability (EDoS) mitigation approaches in cloud: Analysis and open challenges. *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, 206–211. https://doi.org/10.1109/ICECOS.2017.8167135

Dewi, E. K. (2017). Analisis Log Snort Menggunakan Network Forensic. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *2*(2). https://doi.org/10.29100/jipi.v2i2.370

Irina, F. (2017). *Metode Penelitian Terapan.* (1st ed.). Parama Ilmu.

Iskandar. (2020). *Indonesia Dibombardir 88,4 Juta Serangan Siber, Ini Detailnya*. Liputan6.Com. https://www.liputan6.com/tekno/read/4235211/indonesia-dibombardir-884-juta-serangan-siber-ini-detailnya

Khaerani, I., & Handoko, B. (2015). Implementasi dan Analisa Hasil Data Mining untuk Klasifikasi Serangan pada Intrusion Detection System (IDS) dengan Algoritma C4. 5. *Techno.COM*, *14*(3), 181–188. https://doi.org/10.33633/tc.v14i3.943

Kumar, D. A. (2017). Intrusion Detection Systems: A Review. *International Journal of Advanced Research in Computer Science*, *8*(8), 356–370. https://doi.org/10.26483/ijarcs.v8i8.4703

Lazzez, A. (2013). A Survey about Network Forensics Tools. *International Journal of Computer and Information Technology*, *2*(1), 2279–2764.

Muhammad, A. W. (2016). Analisis Statistik Log Jaringan untuk Deteksi Serangan DDOS Berbasis Neural Network. *ILKOM Jurnal Ilmiah*, *8*(3), 220–225. https://doi.org/10.33096/ilkom.v8i3.76.220-225

Paramitha, I. A. S. D., Sasmita, G. M. A., & Raharja, I. M. S. (2020). Analisis Data Log IDS Snort dengan Algoritma Clustering Fuzzy C-Means. *Majalah Ilmiah Teknologi Elektro*, *19*(1), 95. https://doi.org/10.24843/MITE.2020.v19i01.P14

Pilli, E. S., Joshi, R., & Niyogi, R. (2010). A Generic Framework for Network Forensics. *International Journal of Computer Applications*, *1*(11), 1–6. https://doi.org/10.5120/251-408

Purba, W. W., & Efendi, R. (2021). Perancangan dan analisis sistem keamanan jaringan komputer menggunakan SNORT. *AITI*, *17*(2), 143–158. https://doi.org/10.24246/aiti.v17i2.143-158

Sandi, D. V., & Arrofiq, M. (2018). Implementasi Analisis NIDS Berbasis Snort Dengan Metode Fuzy Untuk Mengatasi Serangan LoRaWAN. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *2*(3), 685–696. https://doi.org/10.29207/resti.v2i3.504

Singh, R. R., & Tomar, D. S. (2015). Network Forensics: Detection and Analysis of Stealth Port Scanning Attack. *International Journal of Computer Networks and Communications Security*, *3*(2), 33–42.

Suhartono, S., & Patta, A. R. (2017). Sistem Pengamanan Jaringan Admin Server Dengan Metode Intrusion Detection System (IDS) Snort Menggunakan Sistem Operasi ClearOS. *Jurnal Teknologi Elekterika*, *14*(2), 145. https://doi.org/10.31963/elekterika.v14i2.1220

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 1, JANUARI, 2022, Pp. 56 – 67
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Algoritma K-Nearest Neighbor untuk Memprediksi Prestasi Mahasiswa Berdasarkan Latar Belakang Pendidikan dan Ekonomi

Daru Prasetyawan [(1)*], Rahmadhan Gatra [(2)]
Pusat Teknologi Informasi dan Pangkalan Data (PTIPD), UIN Sunan Kalijaga, Yogyakarta
e-mail : {daru.prasetyawan,rahmadhan.gatra}@uin-suka.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 18 Oktober 2021, direvisi 11 Januari 2022, diterima 12 Januari 2022, dan dipublikasikan 25 Januari 2022.

Abstract

Student academic performance is one measure of success in higher education. Prediction of student academic performance is important because it can help in decision-making. K-Nearest Neighbor (K-NN) algorithm is a method that can be used to predict it. Normalization is needed to scale the attribute value, so the data are in a smaller range than the actual data. Feature selection is used to eliminate irrelevant features. Data cleaning from outliers in the dataset aims to delete data that can affect the classification process. In the classification process, the dataset is divided into a training set by 80% and a validation set by 20% using the cross-validation method. The classification model that is formed is tested using data that is separate from the training data and is evaluated using a confusion matrix. As an evaluation, the K-NN model has 95.85% average accuracy, 95.97% average precision, and 95.84% average recall.

Keywords: Academic Performance, K-NN, Pearson Correlation, Classification, Classification

Abstrak

Prestasi akademik mahasiswa merupakan salah satu ukuran keberhasilan perguruan tinggi. Prediksi prestasi menjadi hal yang penting karena dapat membantu dalam pengambilan keputusan. Algoritma *K-Nearest Neighbor* (K-NN) merupakan algoritma yang dapat digunakan untuk memprediksi prestasi mahasiswa. Normalisasi data diperlukan untuk penguraian nilai atribut sehingga berada dalam kisaran yang lebih kecil dari data sebenarnya. Seleksi fitur digunakan untuk mengeliminasi fitur yang tidak relevan. Pembersihan data dari pencilan di dalam *dataset* bertujuan untuk menghapus data yang dapat mengganggu proses klasifikasi. Proses klasifikasi dilakukan dengan *cross-validation* dengan membagi data menjadi data pelatihan sebesar 80% dan data uji sebesar 20% secara bergantian sebanyak 5 *fold*. Metode *Euclidean*, *Manhattan*, dan *Minkowski* digunakan untuk mengukur jarak di antara dua data. Model klasifikasi yang terbentuk diuji menggunakan data yang terpisah dari data latih dan dievaluasi menggunakan *confusion matrix*. Sebagai hasil evaluasi, diperoleh rata-rata akurasi 95,85%, rata-rata presisi 95,97%, dan rata-rata *recall* 95,84%.

Kata Kunci: Prestasi Akademik, K-NN, Korelasi Pearson, Klasifikasi, Prediksi

1. PENDAHULUAN

Salah satu tujuan perguruan tinggi adalah menciptakan lulusan yang memiliki kemampuan SDM yang unggul. Indeks Prestasi Kumulatif (IPK) merupakan ukuran kemampuan atau prestasi akademik mahasiswa dalam periode tertentu yang dihitung berdasarkan SKS yang telah diambil. Informasi mengenai prediksi prestasi akademik mahasiswa dapat memberi gambaran apakah mahasiswa tersebut akan berhasil memperoleh prestasi akademik yang diharapkan atau tidak. Apabila prestasi akademik seorang mahasiswa dapat diketahui sebelumnya, bahkan pada saat proses seleksi, tentunya dapat membantu perguruan tinggi dalam mengambil keputusan. Pada saat seleksi calon mahasiswa, informasi ini dapat menjadi bahan pertimbangan dalam memutuskan apakah diterima atau tidak. Informasi dan pengetahuan untuk mendukung proses bisnis sebuah perguruan tinggi semakin sangat diperlukan. Ditambah lagi dengan data yang dimiliki sudah sangat besar, tentunya dapat memotivasi untuk mengubah data tersebut menjadi informasi dan pengetahuan yang berguna dengan melakukan mengekstraksi atau menambang

pengetahuan dari data tesebut. Informasi dan pengetahuan tersebut dapat digali dengan memanfaatkan teknologi informasi, khususnya dibidang kecerdasan buatan.

Kecerdasan buatan (*artificial intelligence*) saat ini telah menjadi perhatian lebih karena sudah sangat berpengaruh dalam kehidupan manusia. Kecerdasan Buatan (AI) adalah suatu mesin, program komputer, dan sistem untuk melakukan fungsi intelektual dan kreatif dari seseorang yang secara mandiri menemukan cara untuk memecahkan masalah, mampu menarik kesimpulan dan mengambil keputusan (Rupesh & Choudaiah, 2019). Tujuan utama kecerdasan buatan adalah untuk membuat mesin menjadi lebih pintar sehingga mesin menjadi lebih bermanfaat.

Pembelajaran mesin merupakan cabang kecerdasan buatan berdasarkan ide bahwa sistem dapat belajar dari data, mengidentifikasi pola, dan membuat keputusan dengan intervensi manusia yang minimal. Tujuan dari pembelajaran mesin adalah untuk belajar dari data (Dey, 2016). Pembelajaran mesin menyediakan kemampuan pada sistem untuk secara otomatis belajar dari pengalaman tanpa diprogram secara eksplisit untuk meningkatkan kemampuannya. Proses pembelajaran dimulai dengan melakukan pengamatan atau sekumpulan data, kemudian melatih mesin (komputer) dengan membangun model pembelajaran mesin menggunakan data dan algoritma tertentu untuk mencari pola dalam data tersebut dan membuat keputusan yang lebih baik di masa depan. Tujuan utamanya adalah untuk memungkinkan sebuah komputer dapat belajar secara otomatis tanpa campur tangan atau bantuan manusia dan menyesuaikan tindakan yang sesuai. Pembelajaran mesin sangat berbeda dengan pemrograman secara tradisional. Pada pemrograman tradisional, program dibuat oleh manusia dengan algoritma tertentu, kemudian diberikan input data dan menghasilkan output berdasarkan aturan-aturan yang ada di dalam algoritma tertentu. Sedangkan pada pembelajaran, data dijalankan pada sebuah mesin untuk melakukan pelatihan, kemudian mesin akan membuat programnya sendiri yang dapat dievaluasi pada saat pengujian. Rekayasa perangkat lunak tradisional menggabungkan data dan aturan yang dibuat manusia untuk menciptakan jawaban atas suatu masalah. Sedangkan pembelajaran mesin menggunakan data dan jawaban untuk menemukan aturan di balik suatu masalah (Chollet, 2017).

Secara umum pembelajaran mesin dibagi menjadi pembelajaran terawasi (*supervised learning*), pembelajaran tak terawasi (*unsupervised learning*), dan pembelajaran penguatan (*reinforcement learning*). Pembelajaran terawasi melibatkan penggunaan model untuk mempelajari pemetaan antara input dan variabel target. Pembelajaran terawasi bertujuan untuk mempelajari pemetaan atau aturan antara serangkaian *input* dan *output*. Pembelajaran terawasi membangun model yang membuat prediksi berdasarkan bukti di hadapan ketidakpastian. Tujuan pembelajaran terawasi adalah untuk membangun model dari distribusi label kelas dalam hal fitur prediktor. Model yang dihasilkan kemudian digunakan untuk menetapkan label kelas dari data yang belum diketahui label kelasnya (Reddy & Babu, 2018).

Klasifikasi merupakan pembalajaran terawasi yang memprediksi label dari suatu kelas. Klasifikasi digunakan untuk menemukan model yang menjelaskan atau membedakan kelas data yang dapat memperkirakan kelas dari suatu data yang kelasnya belum diketahui. Klasifikasi adalah proses menemukan kumpulan pola atau fungsi-fungsi yang mendeskripsikan dan memisahkan kelas data satu dengan lainnya, dapat digunakan untuk memprediksi data yang belum memiliki kelas data tertentu. Klasifikasi adalah proses menggunakan model untuk memprediksi nilai yang tidak diketahui (variabel *output*), menggunakan sejumlah nilai yang diketahui (variabel *input*) (Muhammad & Yan, 2015). Algoritma klasifikasi yang sering digunakan di dalam *data mining* antara lain pohon keputusan (*decision tree*), K-Nearest Neighbors (K-NN), dan jaringan syaraf tiruan (*artificial neural network*).

Algoritma K-Nearest Neighbors (K-NN) merupakan algoritma yang sederhana dan mudah diterapkan dalam pembelajaran mesin yang dapat digunakan untuk memecahkan masalah klasifikasi dan regresi. K-NN merupakan algoritma pembelajaran mesin terawasi (*supervised machine learning*), yaitu algoritma yang mengandalkan data input berlabel untuk mempelajari fungsi yang menghasilkan output yang sesuai ketika diberi data baru tanpa label. K-Nearest

Neighbor (KNN) adalah metode yang diterapkan dalam mengklasifikasikan objek berdasarkan data pembelajaran yang paling dekat dengan objek berdasarkan perbandingan antara data sebelumnya dan saat ini (Lubis et al., 2020).

Salah satu metode *data mining* adalah metode klasifikasi dengan *decission tree*. Metode ini dapat digunakan untuk memprediksi prestasi akademik mahasiswa dengan menggunakan algoritma C4.5. Dengan metode klasifikasi ini, hasil pendidikan masa lampau merupakan variabel yang paling menentukan berhasil atau tidaknya seseorang dalam prestasi (Sabna & Muhardi, 2016). Penelitian tentang prediksi prestasi akademik mahasiswa dengan judul "Prediksi Prestasi Akademik Mahasiswa menggunakan Algoritma Random Forest dan C4.5" menggunakan data jenjang pendidikan, program studi, asal daerah, jenis kelamin, SKS semester sebelumnya, dan IP semester sebelumnya, serta membagi kelas prediksi menjadi 3, yaitu IPK Tinggi, IPK Sedang, dan IPK Rendah. Penelitian tersebut menghasilkan akurasi sebesar 87,1% menggunakan algoritma C4.5 dan 92,4% menggunakan algoritma Random Forest (Linawati et al., 2020). Penelitian dengan judul "*Data Mining* untuk Memprediksi Prestasi Siswa Berdasarkan Sosial ekonomi, Motivasi, Kedisiplinan, dan Prestasi Masa Lalu" bertujuan membuat prediksi prestasi belajar siswa berdasarkan status sosial ekonomi orang tua, motivasi, kedisiplinan siswa, dan prestasi masa lalu menggunakan metode *data mining* dengan algoritma J48 yang menghasilkan akurasi sebesar 95,7% (Susanto dan Sudiyatno, 2014). Algoritma K-NN juga telah digunakan dalam memprediksi tingkat kelulusan siswa dengan menggunakan 104 data siswa kelas VI di sekolah swasta area Bekasi (Purwaningsih & Nurelasari, 2021). Penelitian ini menggunakan Nilai PAS, Nilai US Teori, Nilai US Praktik, Nilai UTS, Nilai Prilaku Siswa sebagai atribut dan menghasilkan akurasi sebesar 96,49%. Penelitian berjudul "Algoritma *K-Nearest Neighbor Classification* Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa" juga memanfaatkan Algoritma K-NN untuk memprediksi prestasi mahasiswa (Mustakim & Oktaviani, 2016). Penelitian tersebut menggunakan data Mahasiswa Program Studi Sistem Informasi UIN Sultan Syarif Kasim Riau dengan atribut Jenis Kelamin, Jenis Tinggal, Umur, Jumlah Satuan Kredit Semester (SKS), dan Jumlah Nilai Mutu (NM). Penelitian tersebut membagi menjadi 4 kelas prediksi (Pujian, Sangat Memuaskan, Memuaskan, dan Kurang Memuaskan) dan menghasilkan akurasi sebesar 82%.

Pada Penelitian ini akan dikembangkan sebuah model *data mining* menggunkan algoritma K-Nearest Neigbor (K-NN). Alasannya karena K-NN memiliki kelebihan lebih efektif didata *training* yang besar dan dapat menghasilkan data yang lebih akurat. Selain itu, penelitian ini akan menggunakan data mengenai latar belakang pendidikan dan ekonomi calon mahasiswa pada saat proses pendaftaran dan seleksi penerimaan mahasiswa baru, sehingga informasi yang dihasilkan juga dapat dimanfaatkan dalam proses seleksi tersebut.

2. METODE PENELITIAN

2.1 Data dan Sumber Data

Penelitian ini menggunakan data profil mahasiswa yang terkait dengan latar belakang calon mahasiswa. Untuk mengumpulkan data, penelitian ini menggunakan metode studi pustaka dan observasi. Metode observasi merupakan metode pengumpulan data dengan melakukan pengamatan langsung pada objek yang akan diteliti. Calon mahasiswa baru diminta untuk mengisikan data ekonomi dan pendidikan pada aplikasi yang disediakan.

Dalam penelitian pada suatu populasi, biasanya menggunakan sampel untuk mewakili populasi tersebut dikarenakan akan memakan waktu yang lama dan biaya yang besar apabila menggunakan populasi keseluruhan. Agar sampel yang digunakan dapat mewakili populasinya, diperlukan suatu standar atau teknik dalam mengambil sampel tersebut. Terdapat banyak teknik yang sering digunakan untuk menentukan jumlah sampel dalam suatu populasi. Salah satu teknik teknik pengambilan sampel yang sering digunakan adalah dengan rumus Slovin. Formula rumus Slovin dapat dilihat pada Pers. (1).

$$s = \frac{N}{1+N.e^2} \quad (1)$$

Di mana s adalah sumlah sampel yang akan dihitung, N adalah jumlah populasi, dan e adalah batas toleransi *error*.

Penelitian ini mengambil sampel dari populasi mahasiswa S1 di UIN Sunan Kalijaga yang berjumlah kurang lebih 18.000 mahasiswa. Dengan menggunakan rumus Slovin, diperoleh jumlah sampel minimal yang harus digunakan dengan tingkat toleransi kesalahan 3% adalah 1.047.

2.2 Analisis Data

Tahap ini juga akan memastikan integritas data sehingga tidak menimbulkan masalah pada proses pelatihan. Data latar belakang mahasiswa yang digunakan sejumlah 1.834 data yang tebagi menjadi 4 kelas, yaitu pujian, sangat memuaskan, memuaskan, dan kurang memuaskan. Pada tahap ini dilakukan seleksi data dengan menyeleksi atribut apa saja yang diperlukan. Dalam *dataset* terdapat 6 atribut, yaitu atribut nilai SMA, nilai akreditasi sekolah, penghasilan keluarga, hutang keluarga, jumlah anggota keluarga, dan Indeks Harga Konsumen (IHK) daerah.

Dari Tabel 1 diketahui bahwa jumlah data keseluruhan yang digunakan adalah 1.050 data. Atribut nilai SMA memiliki nilai minimum 45 dan maksimum 90. Atribut nilai akreditasi memiliki nilai minimum 69 dan maksimum 98,99. Atribut penghasilan keluarga memiliki nilai minimum 500.000 dan maksimum 20.000.000. Atribut hutang memiliki nilai minimum 0 dan maksimum 20.000.000. Atribut anggota keluarga memiliki nilai minimum 2 dan nilai maksimum 10. Atribut IHK memiliki nilai minimum 126,45 dan nilai maksimum 140,66. Nilai minimum dan maksimum pada setiap atribut diperlukan dalam proses normalisasi data.

Tabel 1 Deskripsi Data Latar Belakang Pendidikan dan Ekonomi

	Nilai SMA	Nilai Akreditasi	Penghasilan	Hutang	Anggota Keluarga	IHK
count	1050	1050	1050	1050	1050	1050
mean	70,12	90,49	4278310	4168571	4.93	131,91
std	8,03	5,28	3747454	6091702	1.52	3,61
min	45	69	500000	0	2	126,45
25%	65	88	1500000	0	4	129,13
50%	70	91,01	3000000	0	5	130,76
75%	75	94	5500000	8000000	6	133,27
max	90	98,99	20000000	20000000	10	140,66

2.3 Normalisasi Data

Normalisasi adalah proses menyamakan rentang nilai pada setiap atribut dengan skala tertentu (Nasution et al., 2019). Normalisasi diperlukan ketika ada perbedaan besar di dalam dalam rentang fitur yang berbeda. Normalisasi data di dalam paper ini menggunakan teknik *min-max normalization*, yaitu dengan mentransformasikan data ke dalam *range* antara 0 dan 1. Teknik yang menjaga hubungan antara data asli disebut *min-max normalization*. Normalisasi *min-max* adalah teknik sederhana di mana teknik tersebut dapat sesuai dengan batas data yang telah ditentukan sebelumnya (Patro & Sahu, 2015). *Min-max normalization* sering dikenal dengan penskalaan numerik fitur data. Untuk menghitung nilai normalisasi dari anggota dari himpunan nilai-nilai x yang diamati, digunakan formula seperti pada Pers. (2).

$$y = \frac{x-min}{max-min} \quad (2)$$

Di mana *y* adalah data hasil mormalisasi, *x* adalah data yang akan dinormalisasi, *min* adalah nilai data terkecil, dan *max* adalah nilai data terbesar. Dengan persamaan tersebut berarti nilai terkecil

dari suatu variabel ditransformasikan menjadi 0 dan nilai terbesar ditransformasikan menjadi 1, sehingga akan menghasilkan *range* antara 0 sampai dengan 1.

2.4 Seleksi Fitur

Dataset biasanya memiliki variabel atau fitur yang tidak relevan, sehingga dapat mempengaruhi akurasi klasifikasi. Oleh karena itu fitur-fitur tersebut harus dikeluarkan dari *dataset*. Selain itu, semakin sedikit fitur yang digunakan akan mengakibatkan proses klasifikasi menjadi lebih cepat. Pemilihan fitur, sebagai strategi pra-pemrosesan data, telah terbukti efektif dan efisien dalam menyiapkan data berdimensi tinggi untuk masalah *data mining* dan pembelajaran mesin (Li et al., 2018). Penelitian ini menggunakan *correlation-matrix* dalam proses seleksi fitur. *Correlation-matrix* adalah sebuah tabel yang menunjukan koefisian korelasi antar variabel. Analisis korelasi digunakan untuk mengetahui kekuatan antara hubungan korelasi kedua variabel di mana variabel lainnya dianggap berpengaruh dikendalikan atau dibuat tetap (sebagai variabel kontrol) (Romadloni & Hilman F Pardede, 2019). Korelasi antar variabel dihitung dengan *Pearson Correlation* seperti pada Pers. (3).

$$r_{xy} = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2 \sum(y_i-\bar{y})^2}} \qquad (3)$$

Di mana r_{xy} adalah korelasi antara *x* dan *y,* x_i adalah nilai variabel *x* ke-i, $\bar{x}$ adalah rata-rata pada variabel *x,* y_i adalah nilai variabel *y* ke-*i,* dan $\bar{y}$ adalah rata-rata pada variabel *y*.

Pers. (3) digunakan untuk menghitung korelasi antar fitur. Fitur-fitur yang akan digunakan dalam klasifikasi akan dihitung dengan variabel target, yaitu predikat kelulusan. Fitur-fitur yang memiliki nilai korelasi dibawah ambang batas yang ditentukan akan dikeluarkan dari *dataset*. Dalam penelitian ini menggunakan nilai ambang batas 0,4 dalam skala 0 sampai 1, sehingga fitur-fitur yang memiliki nilai korelasi dengan variabel target dibawah nilai ambang batas tidak digunakan dalam proses klasifikasi.

2.5 Pembersihan Data

Salah satu hal yang penting dalam tahap pra pemrosesan data adalah memastikan bahwa data tersebut bersih dari data yang menyimpang sangat jauh dari data lainnya atau yang sering disebut dengan pencilan (*outlier*). Dalam statistika, pencilan adalah titik pengamatan yang jauh dari pengamatan lain. Pencilan dalam suatu kumpulan data merupakan tanda bahwa terjadi ketidaknormalan atau kesalahan pengukuran dalam pengambilan data. Hal ini dapat terjadi karena kesalahan dalam pencatatan atau ketidaktelitian dalam pengumpulan data. Pencilan dalam kumpulan data harus dihilangkan karena dapat mempengaruhi hasil penelitian. Penghitungan *z-score* merupakan salah satu cara yang digunakan untuk mendeteksi pencilan dalam kumpulan data. *Z-score* menggambarkan posisi skor mentah dalam hal jarak dari rata-rata, ketika diukur dalam satuan standar deviasi. Metode *z-score* merupakan teknik yang merepresentasikan perilaku sebuah data dalam kaitannya dengan standar deviasi dan rata-rata dari kumpulan argumen/data (Anusha et al., 2019). *Z-score* diformulasikan seperti pada Pers. (4).

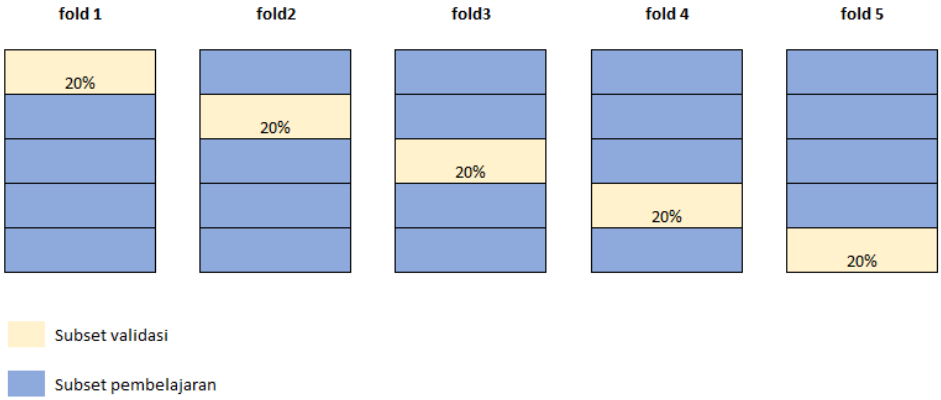
$$z = \frac{x-\bar{x}}{\sigma} \qquad (4)$$

Di mana Z adalah korelasi antara x dan y, x adalah nilai yang akan diukur, $\bar{x}$ adalah rata-rata pada variabel x, dan σ adalah standar deviasi.

Pers. (4) digunakan untuk menghitung nilai z. Data dengan nilai z di atas ambang batas akan dikeluarkan dari *dataset*. Nilai ambang batas yang digunakan dalam penelitian ini adalah 0,4 dari skala 0 sampai 1.

2.6 Klasifikasi

Sebelum melakukan klasifikasi, *dataset* perlu dibagi menjadi data latih dan data uji. Penelitian ini mengambil 20% data sebagai data uji. Dalam menentukan data latih dan data uji, penelitian ini memanfaatkan metode *cross-validation* (CV). *Cross-validation* adalah metode statistik yang dapat digunakan untuk mengevaluasi kinerja model atau algoritma dengan memisahkan data menjadi dua *subset* yaitu *subset* pembelajaran dan *subset* validasi/uji. Model atau algoritma dilatih oleh *subset* pembelajaran dan divalidasi oleh *subset* validasi. *Dataset* akan dibagi menjadi 5 partisi, 4 partisi sebagai subset pembelajaran, dan sisanya sebagai validasi. Proses pembagian antara *subset* pembelajaran dan *subset* validasi akan terus dilakukan sehingga semua data akan berperan sebagai *subset* pembelajaran dan *subset* validasi secara bergantian.



Gambar 1 Pembagian Data Pembelajaran dan Validasi

Penelitian ini akan mencoba menggunakan metode *Euclidean*, *Manhattan*, dan *Minkowski* dalam melakukan perhitungan jarak. Metode dengan akurasi terbaik yang akan digunakan.

2.7 Evaluasi dan Pengujian

Di dalam model klasifikasi, kinerja model yang dihasilkan menggambarkan sejauh mana model tersebut dapat mengklasifikasikan suatu data ke dalam kelas-kelas tertentu. Salah satu metode yang dapat digunakan untuk mengukur kinerja model tersebut adalah *confusion matrix*. Berdasarkan jumlah kelasnya, sistem klasifikasi terbagi menjadi *binary classification* yang hanya mempuanya 2 kelas, dan *multi-class classification* yang memiliki lebih dari 2 kelas). Akurasi merupakan suatu cara yang biasa digunakan untuk mengukur kinerja sistem klasifikasi. Perhitungan akurasi bertujuan untuk memperkirakan seberapa efektif algoritma tersebut dengan menunjukkan probabilitas nilai sebenarnya (*actual*) dan keseluruhan label kelas. dengan kata lain akurasi menilai keefektifan algoritma secara keseluruhan (Sokolova & Lapalme, 2009). Akurasi didefinisikan melalui Pers. (5).

$$Average\ Accuracy\ = \frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fp_i+fn_i+tn_i}}{l} \qquad (5)$$

Di mana *tp* (*true positive*) merupakan jumlah data positif yang diklasifikasikan benar, *tn* (*true negative*) merupakan jumlah data negatif yang diklasifikasikan benar, *fp* (*false positive*) merupakan jumlah data positif yang diklasifikasikan salah, dan *fn* (*false negative*) merupakan jumlah data negatif yang diklasifikasikan salah.

Selain akurasi, presisi dan *recall* juga sering digunakan untuk mengukur kinerja model klasifikasi. *Precision* atau *positive prediction value* merupakan tingkat ketepatan sistem klasifikasi dalam memberikan nilai suatu prediksi. *Precision* mengacu pada persentase hasil klasifikasi yang relevan, sedangkan *recall* mengacu pada persentase total hasil yang relevan yang diklasifikasikan dengan tepat oleh suatu algoritma. *Precision* dihitung dengan membagi jumlah

data positif yang terklasifikasi benar dengan jumlah keseluruhan data yang positif, seperti pada Pers. (6) (Sokolova & Lapalme, 2009).

$$Precision = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp+fp)} \quad (6)$$

Recall dapat didefinisikan sebagai rasio dari jumlah total sampel positif yang terklasifikasi benar dibagi dengan jumlah total sampel positif, sebagaimana dalam Pers. (7).

$$Recall = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp+fp)} \quad (7)$$

3. HASIL DAN PEMBAHASAN

3.1 Normalisasi

Min-max normalization digunakan untuk normalisasi data dengan mentransformasikan data ke dalam *range* antara 0 dan 1. Jika diketahui contoh sampel data x dengan deskripsi:

Nilai SMA (x_1) : 7,33 (*min*: 45; *max*: 90)
Nilai Akreditasi (x_2) : 94 (*min*: 69; *max*: 98,99)
Penghasilan (x_3) : 1000000 (*min*: 500000; *max*: 20000000)
Hutang (x_4) : 0 (*min*: 0; *max*: 20000000)
Anggota keluarga (x_5) : 4 (*min*: 2; *max*: 10)
IHK (x_6) : 130,76 (*min*: 126,45; *max*: 140,66),

dapat dinormalisasikan dengan perhitungan sebagai berikut:

$$y_{(nilai\ SMA)} = \frac{x_{(nilai\ SMA)} - min_{(nilai\ SMA)}}{max_{(nilai\ SMA)} - min_{(nilai\ SMA)}} = \frac{73{,}33-45}{90-45} = 0{,}6295$$

$$y_{(akreditasi)} = \frac{x_{(akreditasi)} - min_{(akreditasi)}}{max_{(akreditasi)} - min_{(akreditasi)}} = \frac{94-69}{98{,}99-69} = 0{,}8336$$

$$y_{(penghasilan)} = \frac{x_{(penghasilan)} - min_{(penghasilan)}}{max_{(penghasilan)} - min_{(penghasilan)}} = \frac{1000000-500000}{20000000-500000} = 0{,}0256$$

$$y_{(hutang)} = \frac{x_{(hutang)} - min_{(hutang)}}{max_{(hutang)} - min_{(hutang)}} = \frac{0-0}{20000000-0} = 0$$

$$y_{(ang,keluarga)} = \frac{x_{(ang,keluarga)} - min_{(ang,keluarga)}}{max_{(ang,keluarga)} - min_{(ang,keluarga)}} = \frac{4-2}{10-2} = 0{,}25$$

$$y_{(ihk)} = \frac{x_{(ihk)} - min_{(ihk)}}{max_{(ihk)} - min_{(ihk)}} = \frac{130{,}76-126{,}45}{140{,}66-126{,}45} = 0{,}3054$$

Hasil dari normalisasi data selanjutnya akan digunakan dalam proses seleksi fitur.

3.2 Seleksi Fitur

Proses seleksi fitur akan mengeluarkan variabel-variabel yang tidak relevan dengan prestasi. Teknik yang digunakan dalam penelitian ini adalah menghitung korelasi linear setiap variabel klasifikasi/fitur dengan variabel target (predikat kelulusan).

$$r_{(nilai\_sma,\ predikat)} = \frac{\sum(nilai\_sma - \overline{nilai\_sma})(predikat_i - \overline{predikat})}{\sqrt{\sum(nilai\_sma_i - \overline{nilai\_sma})^2\ \sum(predikat_i - \overline{predikat})^2}} = \frac{35{,}0749}{55{,}3118} = 0{,}6341$$

$$r_{(akreditasi,\ predikat)} = \frac{\sum(akreditasi_i - \overline{akreditasi})(predikat_i - \overline{predikat})}{\sqrt{\sum(akreditasi_i - \overline{akreditasi})^2\ \sum(predikat_i - \overline{predikat})^2}} = \frac{27{,}578640}{54{,}4063} = 0{,}5069$$

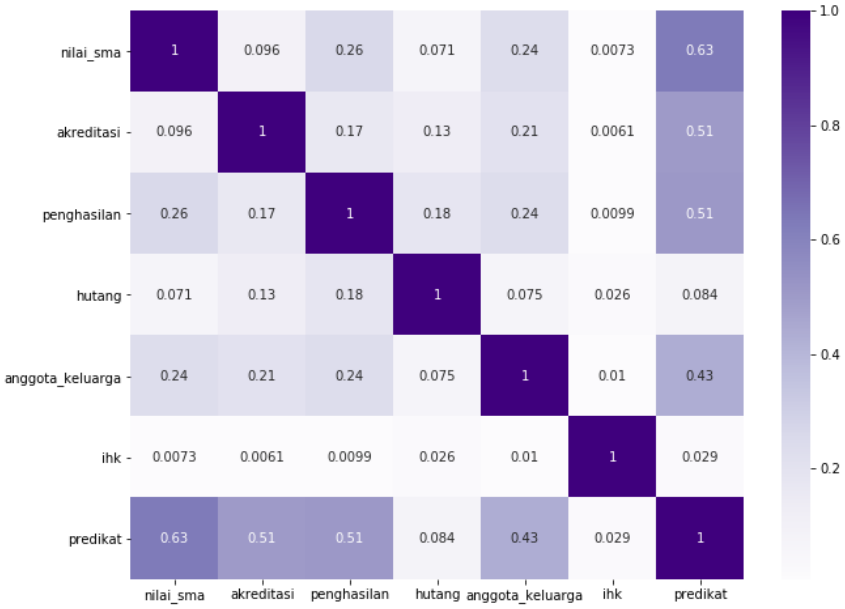
$$r_{(penghasilan,\ predikat)} = \frac{\sum(penghasilan_i - \overline{penghasilan})(predikat_i - \overline{predikat})}{\sqrt{\sum(penghasilan - \overline{penghasilan})^2\ \sum(predikat_i - \overline{predikat})^2}} = \frac{28{,}5776}{55{,}9087} = 0{,}5111$$

$$r_{(hutang,\ predikat)} = \frac{\sum(hutang_i - \overline{hutang})(predikat_i - \overline{predikat})}{\sqrt{\sum(hutang - \overline{hutang})^2\ \sum(predikat_i - \overline{predikat})^2}} = \frac{7{,}7944}{93{,}1918} = 0{,}0836$$

$$r_{(ang,keluarga,\ predikat)} = \frac{\sum(ag,keluarga_i - \overline{ag,keluarga})(predikat_i - \overline{predikat})}{\sqrt{\sum(ag,keluarga - \overline{ag,keluarga})^2\ \sum(predikat_i - \overline{predikat})^2}} = \frac{25{,}2955}{58{,}2462} = 0{,}4343$$

$$r_{(ihk,\ predikat)} = \frac{\sum(ihk_i - \overline{ihk})(predikat_i - \overline{predikat})}{\sqrt{\sum(ihk - \overline{ihk})^2\ \sum(predikat_i - \overline{predikat})^2}} = \frac{2{,}2262}{77{,}7058} = 0{,}0286$$

Variabel target dalam penelitian ini adalah predikat, maka semua variabel dihitung korelasinya dengan variabel predikat. Variabel nilai SMA memiliki nilai korelasi yang paling tinggi, sehingga dapat dikatakan bahwa nilai SMA merupakan variabel yang paling berpengaruh terhadap prestasi mahasiswa. Hasil perhitungan korelasi antar variabel selengkapnya disajikan dalam bentuk matriks korelasi seperti pada Gambar 2.



Gambar 2 Matriks Korelasi Antar Variabel

Dari Gambar 2 diketahui bahwa variabel nilai SMA, akreditasi, penghasilan, dan anggota keluarga berada di atas batas nilai korelasi yang telah ditentukan. Sedangkan variabel hutang dan indeks harga konsumen memiliki nilai korelasi dibawah 0,4. Dengan demikian kedua variabel tersebut tidak digunakan dalam proses pelatihan model, sehingga dalam proses pelatihan model hanya 4 variabel yang akan digunakan, yaitu nilai SMA, akreditasi sekolah, penghasilan keluarga, dan jumlah anggota keluarga.

3.3 Pembersihan Data

Pembersihan data digunakan untuk mengeluarkan data yang menyimpang dari kumpulan data yang akan digunakan dalam proses pelatihan model. Jika diketahui contoh sampel data x (setelah normalisasi) dengan deskripsi:

Nilai SMA (x_1) : 0,63 (*mean*: 0,56; *std*: 0,18)
Nilai Akreditasi (x_2) : 0,83 (*mean*: 0,71; *std*: 0,18)
Penghasilan (x_3) : 0,03 (*mean*: 0,19; *std*:0,18)
Anggota keluarga (x_5) : 0,25 (*mean*: 0,25; *std*: 0,19), perhitungan *z-score* sebagai berikut:

$$Z_{(nilai\ SMA)} = \frac{|x_{(nilai\ SMA)} - mean_{(nilai\ SMA)}|}{std_{(nilai\ SMA)}} = \frac{|0{,}63 - 0{,}56|}{0{,}18} = 0{,}389$$

$$Z_{(akreditasi)} = \frac{|x_{(akreditasi)} - min_{(akreditasi)}|}{std_{(akreditasi)}} = \frac{|0{,}83 - 0{,}71|}{0{,}18} = 0{,}6667$$

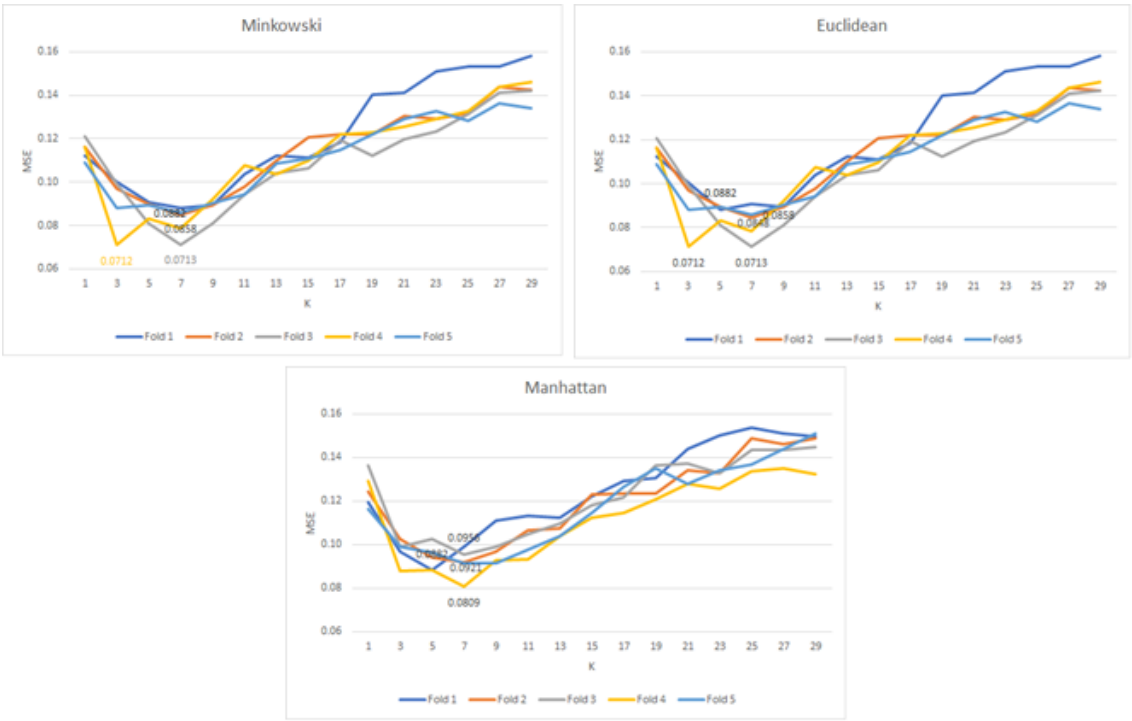
$$Z_{(penghasilan)} = \frac{|x_{(penghasilan)} - min_{(penghasilan)}|}{std_{(penghasilan)}} = \frac{|0{,}03 - 0{,}19|}{0{,}18} = 0{,}8889$$

$$Z_{(ang.keluarga)} = \frac{|x_{(ang.keluarga)} - min_{(ang.keluarga)}|}{std_{(ang.keluarga)}} = \frac{|0{,}25 - 0{,}37|}{0{,}19} = 0{,}6315$$

Dengan menggunakan perhitungan *z-score* diperoleh 14 data dengan nilai *z-score* di luar dari batas yang ditentukan, sehingga data tersebut harus dikeluarkan dari data pelatihan.

3.4 Klasifikasi

Proses klasifikasi dilakukan dengan *cross-validation* dengan membagi data menjadi data pelatihan sebesar 80% dan data uji sebesar 20% secara bergantian sebanyak 5 *fold*. Kemudian dalam proses klasifikasi perhitungan jarak menggunakan metode *Minkowski*, *Euclidean*, dan *Manhattan*. MSE (*Margin Squared Error*) digunakan untuk menentukan nilai k. Jumlah k dengan nilai *error* terendah dipilih untuk melakukan klasifikasi.



Gambar 3 MSE pada Pengukuran Jarak *Minkowski*, *Euclidean*, dan *Manhattan*

Gambar 3 menunjukan grafik nilai MSE untuk setiap *fold* menggunakan metode pengukuran jarak *Minkowski, Euclidean, dan Manhattan*. Pada pengukuran jarak *Minkowski*, nilai MSE terkecil

tarjadi pada *fold-4* dan jumlah k sebanyak 3. Pada pengukuran jarak *Euclidean*, nilai MSE terkecil juga terjadi pada *fold-4* dan jumlah k sebanyak 3. Sedangkan pada pengukuran jarak *Manhattan* nilai MSE terkecil terjadi pada *fold-4* dan jumlah k sebanyak 7.

Tabel 2 Perbandingan Akurasi pada Proses Klasifikasi dengan Perhitungan Jarak *Minkowski, Euclidean, dan Manhattan*

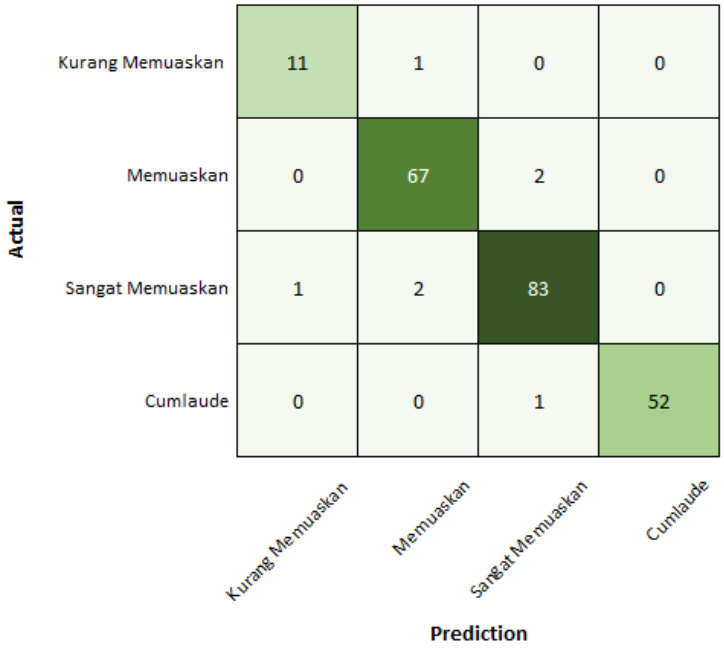
Fold	Minkowski	Euclidean	Manhattan
Fold 1	97,12%	97,12%	94,71%
Fold 2	96,15%	96,15%	96,15%
Fold 3	94,23%	94,23%	95,67%
Fold 4	97,12%	97,12%	96,15%
Fold 5	96,15%	96,15%	95,67%
Rata-rata	**96,15%**	**96,15%**	**95,67%**

Tabel 2 menunjukan perbandingan akurasi saat proses pelatihan menggunakan metode *Minkowski, Euclidean, dan Manhattan.* Akurasi yang sama terjadi pada metode *Minkowski* dan *Euclidean,* yaitu sebesar 96.15%. Sedangkan pada metode *Manhattan,* akurasi yang terjadi lebih rendah dibandingkan *Minkowski* dan *Euclidean*.

3.5 Pengujian dan Evaluasi

Pada tahap pengujian dan evaluasi, sejumlah data uji akan dilakukan klasifikasi dengan menggunakan model klasifikasi yang telah terbentuk. Pengukuran jarak yang digunakan dalam proses klasifikasi adalah metode *Minkowski*, serta jumlah k sebanyak 7. Data uji yang digunakan dalam tahap pengujian pengujian terpisah dari data yang digunakan pada saat pelatihan.

Data uji yang digunakan sebanyak 220 data dengan distribusi data dengan kelas Cumlaude sebanyak 53 data, Sangat Memuaskan sebanyak 86 data, Memuaskan sebanyak 69 data, dan kurang memuaskan sebanyak 12 data. Dalam mengukur kinerja model klasifikasi yang dihasilkan dilakukan dengan *confusion-matrix*. Gambar 6 menunjukan hasil *confusion-matrix* hasil pengujian model klasifikasi.



Gambar 4 *Confusion Matrix* Pengujian Model Klasifikasi

Dari *confusion-matrix* seperti pada Gambar 4 diketahui bahwa pada kelas Cumlaude dengan jumlah sampel sebanyak 53 data terprediksi benar sebanyak 52 data dan hanya 1 data yang terprediksi sebagai kelas Sangat Memuaskan. Pada kelas Sangat Memuaskan dengan jumlah sampel sebanyak 86 terprediksi dengan benar sebanyak 83 data, serta 2 data terprediksi sebagai kelas Memuaskan dan 1 data terprediksi sebagai kelas Kurang memuaskan. Pada kelas Memuaskan dengan jumlah sampel data sebanyak 69 terprediksi secara benar sebanyak 67 data, dan 2 data terprediksi sebagai kelas Sangat memuaskan. Sedangakan pada kelas Kurang Memuaskan dengan jumlah sampel data sebanyak 12 data, 11 data terprediksi dengan benar dan 1 data terprediksi sebagai kelas Memuaskan. Dari *confusion-matrix* tersebut, akurasi model klasifikasi dapat dihitung bersasarkan Pers. (5).

$$Average\ Accuracy = \frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fp_i + fn_i + tn_i}}{l}$$

$$= \frac{\left(\frac{11}{12}\right)+\left(\frac{67}{69}\right)+\left(\frac{83}{86}\right)+\left(\frac{52}{53}\right)}{4}(100\%) = \frac{0.9167+0.9710+0.9651+0.9811}{4}(100\%) = 95{,}85\%$$

TP dan TN di sini adalah sama, yaitu 213 data karena keduanya adalah jumlah dari semua sampel data yang diklasifikasikan benar, terlepas dari kelasnya. FP dengan jumlah 7 merupakan jumlah sampel data yang seharusnya positif tetapi terklasifikasi salah. FN juga dengan jumlah 7 data merupakan jumlah data yang seharusnya bukan anggota dari suatu kelas tetapi terklasifikasi sebagai anggota kelas tersebut. Sehingga akurasi dari model klasifikasi yang terbentuk adalah 96.82 %. Tabel 3 menunjukan kinerja model klasifikasi (presisi, *recall*, dan *f1-score*).

Tabel 3 Kinerja Model Klasifikasi

Kelas	Precision	Recall	F1-Score	Support
Kurang Memuaskan	0,9167	0,9167	0,92	12
Memuaskan	0,9571	0,9710	0,96	69
Sangat Memuaskan	0,9571	0,9651	0,97	86
Cumlaude	1	0,9808	0,99	53
Total/Avg	**0,9597**	**0,9584**	**0,97**	**220**

4. KESIMPULAN

Metode *data mining* khususnya *K-Nearest Neighbor* dapat digunakan dalam memprediksi prestasi mahasiswa. Selain penambahan jumlah data latih, peningkatan kinerja dapat juga dilakukan dengan melakukan pra pemrosesan seperti normalisasi data, seleksi fitur, dan pembersihan pencilan (*outlier*). Latar belakang pendidikan, khususnya nilai SMA merupakan variabel yang paling berpengaruh terhadap prestasi mahasiswa. Kinerja *K-Nearest Neighbor* mencapai akurasi sebesar 95,85%, presisi sebesar 95,97%, dan recall sebesar 95,84% dalam melakukan prediksi prestasi mahasiswa berdasarkan latar belakang pendidikan dan ekonomi. Prediksi prestasi mahasiswa dapat dipengaruhi oleh banyak variabel, sehingga perlu dikembangkan model klasifikasi untuk melakukan prediksi prestasi mahasiswa dengan mempertimbangkan variabel-variabel lain.

DAFTAR PUSTAKA

Anusha, P. V., Anuradha, C., Murty, P. S. R. C., & Kiran, C. S. (2019). Detecting Outliers in High Dimensional Data Sets using Z-Score Methodology. *International Journal of Innovative Technology and Exploring Engineering*, *9*(1), 48–53. https://doi.org/10.35940/ijitee.A3910.119119

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

Dey, A. (2016). Machine Learning Algorithms : A Review. *International Journal of Computer Science and Information Technologies*, *7*(3), 1174–1179.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2018). Feature

Selection. *ACM Computing Surveys*, *50*(6), 1–45. https://doi.org/10.1145/3136625

Linawati, S., Nurdiani, S., Handayani, K., & Latifah. (2020). *Prediksi Prestasi Akademik Mahasiswa Menggunakan Algoritma Random Forest dan C4.5*. *8*(1), 6–13. https://doi.org/10.31294/jki.v8i1.7827

Lubis, A. R., Lubis, M., & Khowarizmi, A.-. (2020). Optimization of distance formula in K-Nearest Neighbor method. *Bulletin of Electrical Engineering and Informatics*, *9*(1), 326–338. https://doi.org/10.11591/eei.v9i1.1464

Muhammad, I., & Yan, Z. (2015). Supervised Machine Learning Approaches: A Survey. *ICTACT Journal on Soft Computing*, *05*(03), 946–952. https://doi.org/10.21917/ijsc.2015.0133

Mustakim, M., & Oktaviani, G. (2016). Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa. *Jurnal Sains, Teknologi, Dan Industri*, *13*(2), 195–202. https://doi.org/10.24014/sitekin.v13i2.1688

Nasution, D. A., Khotimah, H. H., & Chamidah, N. (2019). Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN. *Computer Engineering, Science and System Journal*, *4*(1), 78. https://doi.org/10.24114/cess.v4i1.11458

Patro, S. G. K., & Sahu, K. K. (2015). Normalization: A Preprocessing Stage. *IARJSET*, 20–22. https://doi.org/10.17148/IARJSET.2015.2305

Purwaningsih, E., & Nurelasari, E. (2021). Penerapan K-Nearest Neighbor Untuk Klasifikasi Tingkat Kelulusan Pada Siswa. *Syntax: Jurnal Informatika*, *10*(01), 46–55.

Reddy, R. V. K., & Babu, U. R. (2018). A Review on Classification Techniques in Machine Learning. *International Journal of Advance Research in Science and Engineering*, *7*(3), 40–47.

Romadloni, N. T., & Hilman F Pardede. (2019). Seleksi Fitur Berbasis Pearson Correlation Untuk Optimasi Opinion Mining Review Pelanggan. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *3*(3), 505–510. https://doi.org/10.29207/resti.v3i3.1189

Rupesh, G., & Choudaiah, S. (2019). Artificial Intelligence and its Role in Near Future. *International Journal of Science Research (IJSR)*, *8*(3), 893–898.

Sabna, E., & Muhardi, M. (2016). Penerapan Data Mining Untuk Memprediksi Prestasi Akademik Mahasiswa Berdasarkan Dosen, Motivasi, Kedisiplinan, Ekonomi, dan Hasil Belajar. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer Dan Teknologi Informasi*, *2*(2), 41. https://doi.org/10.24014/coreit.v2i2.2392

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Susanto, H., & Sudiyatno, S. (2014). Data mining untuk memprediksi prestasi siswa berdasarkan sosial ekonomi, motivasi, kedisiplinan dan prestasi masa lalu. *Jurnal Pendidikan Vokasi*, *4*(2), 222–231. https://doi.org/10.21831/jpv.v4i2.2547

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 7, No. 1, JANUARI, 2022, Pp. 68 – 74
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Pengenalan Pola Huruf Hijaiyah dengan Metode *Local Binary Pattern* Menggunakan Algoritma *Fuzzy K-Nearest Neighbor*

Asdar [(1)], Rizal Adi Saputra [(2)*], Ika Purwanti Ningrum [(3)]
Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari
e-mail : {asdar.alkhalil,ika.purwanti.n}@gmail.com, rizaladisaputra@uho.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 2 Agustus 2021, direvisi 22 Oktober 2021, diterima 24 Oktober 2021, dan dipublikasikan 25 Januari 2022.

Abstract

A letter is a form, stroke, or symbol writing system. Any information obtained from a sentence depends on the letters are written clearly. Finding written hijaiyah letters can be recognized by humans, but will be difficult if a computer tries to recognize them. The reason system is difficult is because of the large variety of different letters. This study aims to make it easier for someone to learn to recognize hijaiyah letters by using the Local Binary Pattern method for the feature extraction process. The results of feature extraction will take the maximum value of the histogram of each letter. And results feature extraction will be carried out classification process using the Fuzzy K-Nearest Neighbor algorithm until finally hijaiyah letters can be recognized. Based on experimental results that have been carried out, the highest level of accuracy is obtained when the amount of training data is 154 data and the number of data testing is 29 data, resulting in an accuracy rate of 96.55%.

Keywords: Feature Extraction, Fuzzy K-Nearest Neighbor, Hijaiyah, Histogram, Local Binary Pattern

Abstrak

Huruf adalah sebuah bentuk, goresan, atau lambang dari suatu sistem tulisan. Sebuah informasi yang didapatkan dari suatu kalimat tergantung pada cara penulisan huruf yang jelas. Dalam mengidentifikasi sebuah tulisan huruf hijaiyah dapat dikenali dengan penglihatan seorang manusia namun akan menjadi sulit apabila sebuah komputer yang berusaha untuk mengenalinya. Alasan yang menyebabkan sistem kesulitan adalah karena banyaknya variasi huruf yang berbeda-beda. Penelitian ini bertujuan untuk memudahkan seseorang dalam belajar mengenali tulisan huruf hijaiyah dengan menggunakan metode *Local Binary Patern* untuk proses ekstraksi fiturnya, kemudian hasil dari ekstraksi fitur tersebut akan diambil nilai maksimum dari histogram setiap huruf. Dan hasil dari ekstraksi fitur tersebut akan dilakukan proses klasifikasi menggunakan algoritma *Fuzzy K-Nearest Neighbor* sampai akhirnya huruf hijaiyah dapat dikenali. Berdasarkan hasil uji coba yang telah dilakukan, tingkat akurasi tertinggi diperoleh ketika jumlah data *training* sebanyak 154 data dan jumlah data *testing* sebanyak 29 data, menghasilkan tingkat akurasi sebesar 96,55%.

Kata Kunci: Ekstraksi Fitur, *Fuzzy K-Nearest Neighbor*, Hijaiyah, Histogram, *Local Binary Pattern*

1. PENDAHULUAN

Bagi manusia tentunya tidaklah sulit untuk mengenali sebuah huruf tulisan tangan walaupun berbeda-beda bentuk antara penulis satu dengan penulis lain. Namun hal itu menjadi sulit jika mesin yang berusaha untuk mengenali tulisan tangan dari manusia yang berbeda-beda antara satu dan yang lainnya. Dalam kasus ini lebih sulit jika tulisan tangan yang akan dikenali yaitu tulisan huruf hijaiyah. Saat ini telah banyak penelitian tentang bagaimana melakukan pengenalan/identifikasi tulisan tangan. Banyak pula metode yang digunakan seperti pencocokan citra, algoritma genetika, dan pendekatan sintaktik. Namun metode-metode tersebut kurang efektif untuk mengenali tulisan tangan yang sangat kompleks terutama dari segi bentuk dan ukuran huruf hijaiyah. Berdasarkan latar belakang tersebut peneliti tertarik untuk melakukan

penelitian klasifikasi huruf hijaiyah dengan menggunakan ekstraksi ciri yang dapat membedakan antara huruf satu dan lainnya.

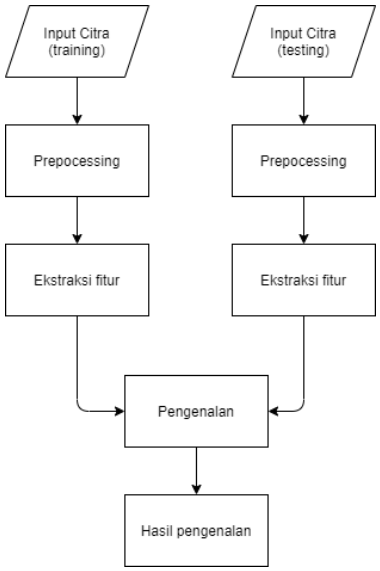
Pengolahan citra digital merupakan salah satu solusi untuk melakukan klasifikasi identifikasi huruf hijaiyah seperti pada penelitian (Sanjaya & Widodo, 2018). Berdasarkan penelitian tersebut pengolahan citra dengan partisi 3x3 memiliki hasil akurasi paling tinggi yakni 84,52% dibandingkan ukuran partisi citra lainnya. Dalam melakukan proses ekstraksi fitur terdapat juga metode lain yang memanfaatkan partisi citra atau metode ketetanggan salah satunya adalah *Local Binary Pattern.* Sebelumnya telah banyak penelitian mengenai pengolahan citra pada pengenalan pola yang menggunakan metode *Local Binary Pattern* seperti pada penelitian Sari & Saputra (2018) membahas pengenalan *finger vein* dan pada penelitian Sanjaya & Setiawan (2018) membahas pengenalan pola huruf latin. Selain itu terdapat pula penelitian Fajriani (2017) yang menggunakan *Local Binary Pattern* untuk melakukan pengenalan pola telapak tangan serta Surrisyad & Yazid (2017) yang mengimplementasikan metode ekstraksi fitur tersebut dalam pengenalan pola huruf pegon jawa. Dari keempat penelitan tersebut rata-rata hasil ekstraksi fitur menunjukkan hasil akurasi di atas 90%.

Pengenalan pola huruf hijaiyah juga sebelumnya telah dilakukan oleh (Faturrahman, 2018) dan (Saputra & Asdar, 2021) dengan menggunakan metode *Backpropagation*. Namun hasil klasifikasi masih belum optimal dikarenakan data *training* yang digunakan masih terbatas yakni masing-masing menggunakan 2 dan 3 data *training*. Selain itu waktu pemrosesan data *training* yang lama menjadi kendala dalam penggunaan metode *Backpropagation*. Untuk mengatasi masalah tersebut, peneliti akan menambah data *training* menjadi 5 dan menggunakan metode klasifikasi *Fuzzy K-Nearest Neighbor* agar waktu pemrosesan data *training* dapat lebih cepat. Penelitian sebelumnya pun telah menunjukkan klasifikasi dengan *Fuzzy K-Nearest Neighbor* memiliki akurasi yang baik seperti pada penelitian (Gafar & Sari, 2018) sebesar 88% dan (Setiyorini & Sari, 2017) sebesar 93%.

Berdasarkan latar belakang tersebut peneliti mengusulkan sebuah penelitian yang berjudul "Pengenalan Pola Huruf Hijaiyah dengan Metode *Local Binary Pattern* dan Menggunakan Algoritma *Fuzzy K-Nearest Neighbor"*.

2. METODE PENELITIAN

Sistem pengenalan pola huruf hijaiyah pada aplikasi ini memiliki beberapa proses seperti yang ditunjukkan pada Gambar 1.



Gambar 1 Gambaran Umum Sistem

2.1 Tahap Pengumpulan Data

Pengumpulan data dilakukan dengan membuat huruf hijaiyah secara manual menggunakan aplikasi desain *inspect* dengan *output* gambar format **jpg*. Data yang digunakan sebanyak 174 data yang terbagi menjadi 145 data *training* dan 29 data *testing.* Masing-masing huruf hijaiyah akan diakusisi sebanyak 6 kali dengan rotasi yang berbeda-beda yakni 0^0,30^0, 45^0, 60^0, 90^0, dan 120^0. Dari keenam citra tersebut 5 citra akan menjadi data *training* dan 1 citra akan menjadi data *testing* untuk setiap huruf hijaiyah. Data *training* digunakan sebagai data latih, sedangkan data *testing* digunkan untuk menguji apakah data yang telah di-*training* menghasilkan luaran yang diinginkan. Gambar 2 menunjukkan citra huruf hijaiyah untuk huruf ba dalam berbagai rotasi.

Gambar 2 Citra Huruf Hijaiyah

2.2 Tahap *Preprocessing*

Pada tahapan ini, yang dilakukan dalam *preprocessing* yaitu konversi RGB ke *grayscale* yang nantinya dari data yang telah dikonversi akan diekstraksi fiturnya. Proses konversi citra digunakan untuk mendapatkan nilai warna yang lebih sederhana. Di mana warna *grayscale* hanya mempunyai intensitas warna 0 - 255 untuk setiap pikselnya.

2.3 Tahap Ekstraksi Fitur

Pada penelitian ini ekstraksi fitur yang digunakan adalah *Local Binary Pattern* (LBP). Operator LBP dan telah terbukti sebagai deskriptor tekstur yang tangguh. Operator LBP melabeli piksel-piksel dari sebuah citra dengan melakukan proses *thresholding* ketetanggaan 3x3 dari masing masing piksel sebagai nilai tengah dan mengubah hasilnya menjadi nilai biner, dan 256-bin LBP histogram digunakan sebagai *texture descriptor* (Nanni et al., 2012). Bilangan biner yang dihasilkan (disebut *Local Binary Pattern* atau LBP *codes*) mengkodekan lokal primitif termasuk variasi dari lengkungan sisi, bintik, area datar, dan lainnya.

Setelah melabeli citra dengan *Local Binary Pattern*. Histogram dari citra *f(x,y)* dapat didefinisikan sebagai,

$$Hi = \sum_{x,y} I(fl(x,y)=i), \quad i=0,\ldots,n-1 \qquad (1)$$

Di mana n adalah jumlah label biner yang dihasilkan oleh LBP operator dan

$$I(A)= \begin{cases} 1 \text{ A is true} \\ 0 \text{ A is false} \end{cases} \qquad (2).$$

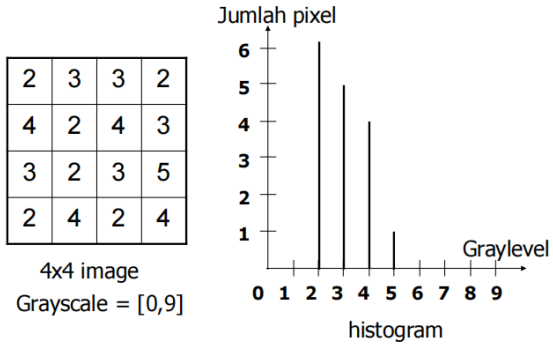
LBP histogram ini mengandung informasi tentang distribusi *local micro-pattern*, seperti sisi-sisi, titik-titik, dan area datar, diseluruh permukaan citra, sehingga secara statistik dapat mendeskripsian karakteristik yang terdapat pada citra (Liu et al., 2017). Untuk mengikutsertakan informasi bentuk dari huruf, ekstraksi LBP dilakukan pada citra yang dibagi sama rata menjadi *region-region* yang lebih kecil *R0,R1,…,Rm*. Histogram yang diekstrak dari tiap-tiap *sub region* di-*concat* menjadi sebuah histogram ciri tertambah secara spasial yang didefinisikan dengan

$$Hi,j = \sum_{x,y} I\{fl(x,y) = i\} I\{x,y\} \epsilon Rj\} \qquad (3)$$

di mana $i = 0, \ldots, n-1, j = 0, \ldots, m-1$.

Histogram citra (*image histogram*) merupakan informasi yang penting mengenai isi citra digital. Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu citra atau bagian tertentu di dalam citra (Wu et al., 2017). Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (*relative*) dari intensitas pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontas (*contrast*) dari sebuah gambar. Karena itu, histogram adalah alat bantu yang berharga dalam pekerjaan pengolahan citra baik secara kualitatif maupun kuantitatif (Kaur & Sohi, 2017).

Menghitung histogram, misalkan sebuah citra mempunyai *L* level nilai keabuan, $0[L-1$. Lalu hitung frekuensi kemunculan setiap nilai keabuan *j* dengan cara menghitung jumlah *pixel* yang mempunyai nilai keabuan tersebut (Riana et al., 2022). Perhitungan ini dilakukan untuk $j = 0, 1, 2, \ldots, L-1$.



Gambar 3 Proses Perhitungan Histogram

Metode LBP digunakan untuk mengekstraksi citra berupa tekstur yang dimana hasil ekstraksinya berupa histogram dan nilai dari histogram itu sendiri. Kemudian nilai maksimum dari histogram tersebut yang akan diolah kedalam proses selanjutnya. Setiap citra akan memiliki nilai maksimum histogram yang selanjutnya menjadi ciri khusus pada masing-masing citra tersebut yang dapat membedakan dengan citra lainnya.

2.4 Tahap Pengenalan

Pada penelitian ini pengenalan yang digunakan adalah algoritma *Fuzzy K-Nearest Neighbor* (FKNN). Konsep dasar dari metode *Fuzzy K-Nearest Neighbor* (FKNN) adalah memberikan derajat keanggotaan sebagai representasi dari jarak *K-Nearest Neighbor* fitur citra dan keanggotaannya pada beberapa kemungkinan kelas (Abu Alfeilat et al., 2019).

Persamaan *μ(x,yi)* adalah nilai keanggotaan data *x* ke kelas *yi*, variabel *k* merupakan jumlah tetangga terdekat yang digunakan. Maka *μ(xj,yi)* merupakan nilai keanggotaan data tetangga dalam *k* tetangga pada kelas *yi* di mana nilainya 1 jika data latih *xj* memiliki kelas *yi* , untuk *d (x,xj)* adalah jarak dari data *x* ke data *xj* dalam *k* tetangga terdekat, *m* merupakan *scalling factor* untuk nilai keanggotaan *μ(x,yi)*. Untuk menghitung *μ(x,yi)* , digunakan Pers. (4).

$$\mu(x,yi) = \frac{\sum_{j=1}^{k} \mu(xj,yi) * d(x-xj)^{-\frac{2}{m-1}}}{\sum_{j=1}^{k} d(x-xj)^{-\frac{2}{m-1}}} \qquad (4)$$

Karena menggunakan metode *Fuzzy K-Nearest Neighbor* (FKNN), setiap elemen dari data uji *x* akan diklasifikasikan ke dalam lebih dari satu kelas dengan nilai keanggotaan *μ(x,yi)*. Namun yang akan diambil sebagai kelas dari elemen *x* adalah kelas *yi* dengan nilai keanggotaan *μ(x,yi)* tertinggi.

Fitur tekstur citra huruf hijaiyah hasil LBP (nilai maksimum histogram) selanjutnya akan digunakan untuk proses pengenalan. Pengenalan citra huruf hijaiyah dilakukan dengan mencocokkan fitur

tekstur huruf hijaiyah pada citra *training* dan fitur tekstur huruf hijaiyah pada citra *testing* menggunakan metode *Fuzzy K-Nearest Neighbor* (FKNN). Untuk mengevaluasi sistem pengenalan huruf hijaiyah yang dibangun, digunakan pengujian identifikasi yaitu setiap dua citra *training* (latih) dicocokkan dengan setiap dua citra *testing* (uji). Kemudian akan dihitung besar akurasi dari pengenalan seluruh citra *training* (latih). Akurasi diperoleh dengan menghitung jumlah dari pengenalan citra data *training* yang benar. Untuk perhitungan akurasi digunakan seperti pada Pers. (5).

$$\text{Akurasi} = \frac{\text{Jumlah benar}}{\text{Total citra uji}} \times 100\% \qquad (5)$$

3. HASIL DAN PEMBAHASAN

Pengujian pada penelitian ini dilakukan untuk melihat seberapa besar akurasi dalam mengenali pola huruf hijaiyah. Data *training* yang digunakan pada penelitian ini sebanyak 145 data dari 5 data huruf pada setiap karakter huruf hijaiahnya dengan rotasi yan beragam dan untuk data *testing* sebanyak 29 data. Pada penelitian ini nilai k optimal yang digunakan pada *Fuzzy K-Nearest Neighbor* adalah $k = 3$. Hal ini berdasarkan pada penelitian terdahulu seperti yang dilakukan Fadholi et al. (2019) untuk pengenalan citra makanan tradisional. Hasil penelitian tersebut menunjukkan nilai $k = 3$ memperoleh hasil persentase lebih tingi dibandingkan $k = 1, 5, 7, dan\ 9$. Pengujian pada penelitian ini akan menggunakan skema perbandingan data *training* : data *testing* = 90 : 10 sehingga nilai k optimal pada *Fuzzy K-Nearest Neighbor* yang paling cocok adalah $k = 3$ seperti pada hasil penelitian (Rustamaji et al., 2019) yang melakukan klasifikasi data *categorial* dengan menggunakan *Fuzzy K-Nearest Neighbor*.

Tabel 1 Hasil Pengujian

No.	Huruf Hijaiah	Hasil
1	ا	Dikenali
2	ب	Dikenali
3	ت	Dikenali
4	ث	Dikenali
5	ج	Dikenali
6	ح	Dikenali
7	خ	Dikenali
8	د	Dikenali
9	ذ	Dikenali
10	ر	Dikenali
11	ز	Dikenali
12	س	Dikenali
13	ش	Dikenali
14	ص	Dikenali
15	ض	Dikenali
16	ط	Dikenali
17	ظ	Dikenali
18	ع	Dikenali
19	غ	Dikenali
20	ف	Tidak dikenali
21	ق	Dikenali
22	ك	Dikenali
23	ل	Dikenali
24	م	Dikenali
26	ن	Dikenali
27	ه	Dikenali
28	و	Dikenali
29	ئ	Dikenali

Tabel 1 merupakan tabel hasil pengujian huruf hijaiyah. Dari tabel tersebut dapat dilihat dari total 29 huruf data *testing* sebanyak 28 huruf dapat dikenali dan bernilai benar dan hanya 1 huruf yang tidak dikenali atau identifikasi salah yakni pada huruf ف. Sehingga akurasi pengujian dapat dihitung dengan menggunakan persamaan (5).

$$Akurasi = \frac{28}{29}\ x\ 100\% = 96.55\%$$

Berdasarkan perhitungan akurasi dapat dikatakan bahwa persentase pengenalan huruf hijaiyah dengan menggunakan *Local Binary Pattern* dan *Fuzzy K-Nearest Neighbor* menghasilkan akurasi 96,55%.

4. KESIMPULAN

Berdasarkan analisis hasil pengujian maka dapat disimpulkan bahwa tingkat keakuratan aplikasi pengenalan pola Huruf Hijaiyah dengan mengimplementasikan metode *Local Binary Pattern* dan menggunakan algoritma *Fuzzy K-Nearest Neighbor* adalah sebesar 96,55%. Metode dan algoritma ini sudah cukup baik untuk mengenali huruf hijaiyah akan tetapi belum bisa mengenali semua huruf hijaiah dengan benar. Hal ini disebabkan oleh hasil ekstraksi fitur pada setiap huruf hijaiyah memiliki nilai yang hampir sama sehingga ketika proses klasifikasi masih terjadi *error* pada satu huruf hijaiyah. Adapun saran untuk penelitian selanjutnya mengenai pengenalan pola huruf hijaiyah yaitu perlu digunakan pengembangan dari *Local Binary Pattern* seperti *Local Line Binary Pattern* serta memfokuskan penelitian pada pencarian nilai k optimal yang digunakan pada *Fuzzy K-Nearest Neighbor* untuk pengenalan huruf hijaiyah.

DAFTAR PUSTAKA

Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. B. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data*, *7*(4), 221–248. https://doi.org/10.1089/big.2018.0175

Fadholi, R., Sari, Y. A., & Bachtiar, F. A. (2019). Pengenalan Citra Makanan Tradisional menggunakan Fitur Hue Saturation Pengenalan Citra Makanan Tradisional menggunakan Fitur Hue Saturation Value dan Fuzzy k-Nearest Neighbor. *Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(7), 6556–6566.

Fajriani, N. (2017). Pengenalan Pola Garis Telapak Tangan Menggunakan Metode Fuzzy K-Nearest Neighbor. *Edutic - Scientific Journal of Informatics Education*, *4*(1). https://doi.org/10.21107/edutic.v4i1.3385

Faturrahman, I. (2018). Pengenalan Pola Huruf Hijaiyah Khat Kufi dengan Metode Deteksi Tepi Sobel Berbasis Jaringan Syaraf Tiruan Backpropagation. *JURNAL TEKNIK INFORMATIKA*, *11*(1), 37–46. https://doi.org/10.15408/jti.v11i1.6262

Gafar, A. A., & Sari, J. Y. (2018). Sistem Pengenalan Bahasa Isyarat Indonesia dengan Menggunakan Metode Fuzzy K-Nearest Neighbor. *Jurnal ULTIMATICS*, *9*(2), 122–128. https://doi.org/10.31937/ti.v9i2.671

Kaur, H., & Sohi, N. (2017). A Study for Applications of Histogram in Image Enhancement. *The International Journal of Engineering and Science*, *06*(06), 59–63. https://doi.org/10.9790/1813-0606015963

Liu, P., Guo, J.-M., Chamnongthai, K., & Prasetyo, H. (2017). Fusion of color histogram and LBP-based features for texture image retrieval and classification. *Information Sciences*, *390*, 95–111. https://doi.org/10.1016/j.ins.2017.01.025

Nanni, L., Lumini, A., & Brahnam, S. (2012). Survey on LBP based texture descriptors for image classification. *Expert Systems with Applications*, *39*(3), 3634–3641. https://doi.org/10.1016/j.eswa.2011.09.054

Riana, D., Syahrani, M., Mandiri, U. N., Melayu, C., & Timur, K. J. (2022). Pengelolaan Citra Digital Dengan Menggunakan Metode Transformasi Gryascale dan Pemerataan Histogram. *Jurnal Teknik Informatika Kaputama (JTIK)*, *6*(1), 108–119.

Rustamaji, H. C., Simanjuntak, O. S., Luhrie, S. F., Yuwono, B., & Juwairiah. (2019). Categorical

Data Classification based on Fuzzy K-Nearest Neighbor Approach. *2019 5th International Conference on Science in Information Technology (ICSITech)*, 171–175. https://doi.org/10.1109/ICSITech46713.2019.8987477

Sanjaya, A., & Setiawan, A. B. (2018). Pengenalan Tulisan Tangan Huruf Latin Dengan Menggunakan Metode K-Nearest Neighbour. *Journal of Chemical Information and Modeling*, *2*(2), 1689–1699.

Sanjaya, A., & Widodo, D. W. (2018). Identifikasi Tulisan Tangan Huruf Hijaiyah. *Network Engineering Research Operation*, *4*(1), 23–29. https://doi.org/10.21107/nero.v4i1.108

Saputra, R. A., & Asdar. (2021). Pengenalan Pola Huruf Arab Dengan Metode Backpropagation. *Proceeding Konferensi Nasional Ilmu Komputer (KONIK)*, *5*, 424–427.

Sari, J. Y., & Saputra, R. A. (2018). Pengenalan Finger Vein Menggunakan Local Line Binary Pattern dan Learning Vector Quantization. *Jurnal ULTIMA Computing*, *9*(2), 52–57. https://doi.org/10.31937/sk.v9i2.790

Setiyorini, A., & Sari, J. Y. (2017). Perbaikan Kualitas Citra Untuk Klasifikasi Daun Menggunakan Metode Fuzzy K-Nearest Neighbor. *Jurnal ULTIMATICS*, *9*(2), 129–135. https://doi.org/10.31937/ti.v9i2.688

Surrisyad, H., & Yazid, A. S. (2018). Aplikasi Jaringan Syaraf Tiruan dalam Pengenalan Pola Huruf Pegon Jawa. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *3*(1), 34. https://doi.org/10.14421/jiska.2018.31-04

Wu, X., Liu, X., Hiramatsu, K., & Kashino, K. (2017). Contrast-accumulated histogram equalization for image enhancement. *2017 IEEE International Conference on Image Processing (ICIP)*, *typically 256*, 3190–3194. https://doi.org/10.1109/ICIP.2017.8296871

9 772527 583007