

Penerapan *Regular Expression* untuk *Data Cleaning* dan *Log Parsing* dalam Pengolahan Data *Syslog*

Riki Ramdani Saputra^{1*}, Viby Ladyscha Yalasena Winarno², Jesika Oktaviani²

¹Badan Riset dan Inovasi Nasional, Jakarta Pusat.

²Program Studi Ilmu Komputer, Sekolah Sains Data, Matematika, dan Informatika, IPB University, Kabupaten Bogor.

Artikel Info

Kata Kunci:

Data Cleaning;
Log Parsing;
Log Server;
Regular Expression;
Syslog.

Keywords:

Data Cleaning;
Log Parsing;
Log Server;
Regular Expression;
Syslog.

Riwayat Artikel:

Submitted: 25 April 2026
Accepted: 17 September 2026
Published: 18 September 2026

Abstrak: Data log pada *Syslog server* berperan penting dalam mendukung monitoring keamanan jaringan, namun umumnya masih tidak terstruktur sehingga mempersulit proses analisis. Penelitian ini bertujuan menerapkan *Regular Expression* (Regex) untuk melakukan *data cleaning* dan *log parsing* pada data *Syslog* agar lebih terstruktur dan mudah dianalisis. Metode yang digunakan meliputi normalisasi atribut *Tag*, *Time*, dan *Host*, ekstraksi informasi dari kolom *Message* menggunakan pola *Common Event Format* (CEF), penyimpanan data ke MySQL, serta pengembangan antarmuka berbasis web. Hasil penelitian menunjukkan bahwa Regex mampu menghasilkan data log yang lebih konsisten dan terstruktur untuk proses analisis. Sistem yang dikembangkan juga berhasil mengelola data melalui fitur pencarian, penyaringan, dan unggah data. Hasil *Black Box Testing* menunjukkan seluruh 5 skenario pengujian berhasil dijalankan dengan tingkat keberhasilan fungsional sebesar 100%. Penelitian ini menunjukkan bahwa penerapan Regex efektif dalam mendukung *data cleaning*, *log parsing*, dan monitoring keamanan jaringan berbasis *Syslog*.

Abstract: *Syslog data play an important role in supporting network security monitoring; however, they are often unstructured and difficult to analyze. This study applies Regular Expression (Regex) for data cleaning and log parsing to improve the structure and usability of Syslog data. The proposed method includes the normalization of Tag, Time, and Host attributes, information extraction from the Message field using the Common Event Format (CEF), storage of processed data in MySQL, and the development of a web-based interface. The results show that Regex produces more consistent and structured log data for further analysis. The developed system also supports data search, filtering, and upload functionalities. Black Box Testing demonstrated that all 5 test cases were successfully executed, achieving a functional success rate of 100%. These findings indicate that Regex is effective for supporting Syslog data cleaning, log parsing, and network security monitoring.*

Corresponding Author:

Riki Ramdani Saputra
Email: riki007@brin.go.id

PENDAHULUAN

Perkembangan teknologi informasi dan ilmu pengetahuan yang cukup pesat mendorong meningkatnya ketergantungan masyarakat terhadap internet. Peningkatan kebutuhan terhadap internet memberikan banyak manfaat dalam berbagai aspek. Namun, penggunaan internet yang tidak bertanggung jawab dapat membuka peluang munculnya dampak negatif seperti pelanggaran dalam mengakses *website* atau konten yang bersifat ilegal. Konten tersebut dikategorikan sebagai *negative content as a code* yang mencakup berbagai bentuk ancaman bagi sistem komputer seperti *computer viruses*, *worms*, *trojan horses*, *ransomware*, *spyware*, *adware*, dan *scareware* (Hamid, 2022). Ketika terjadi serangan, administrator dapat menelusuri log server untuk mengetahui darimana serangan tersebut berasal. Data tersebut dapat diketahui melalui alamat IP dan pesan log yang dipakai ketika mengakses web server (Yogi et al., 2019). Log adalah catatan otomatis dari aktivitas sistem dan pengguna, yang dihasilkan oleh berbagai perangkat dan aplikasi. Log berfungsi penting dalam pemantauan, analisis perilaku, dan keamanan sistem, karena mencatat detail aktivitas yang tidak tercatat di sumber lain (Sarker, 2021). Untuk memperoleh data log server, administrator biasanya memanfaatkan perangkat server khusus seperti *Syslog server*.

Syslog server merupakan sistem yang berfungsi sebagai manajemen log yang dapat menerima pesan kejadian dari berbagai perangkat melalui jaringan IP dan mendukung fungsi penyimpanan log secara terpusat (Anton et al., 2024). *Syslog server* mencatat seluruh aktivitas sistem operasi beserta pesan-pesan yang terkait, seperti peringatan, kesalahan, dan peristiwa kritis, yang berperan penting dalam proses pemantauan serta kepatuhan penggunaan sistem (Sasidharan, 2022). Meskipun data log dari berbagai perangkat jaringan dapat digunakan untuk menilai kondisi keamanan jaringan, jumlahnya yang besar menyebabkan informasi di dalamnya belum dapat dimanfaatkan secara optimal. Sheeraz et al. (2024) menjelaskan bahwa data log dalam jumlah besar dari berbagai sumber perlu dianalisis lebih lanjut agar dapat menghasilkan informasi yang lebih ringkas dan bermakna. Selain itu, Chuah et al. (2025) juga menyatakan bahwa log sistem merupakan salah satu sumber informasi utama bagi analisis keamanan dalam mendeteksi dan memprediksi serangan siber, terutama ketika pengguna mengakses situs ilegal. Berdasarkan permasalahan tersebut, penelitian terkait analisis data log server menjadi penting dilakukan sebagai upaya untuk meningkatkan keamanan sistem dan mendeteksi ancaman siber secara lebih dini.

Salah satu teknik yang dapat diterapkan dalam proses analisis tersebut adalah penggunaan *Regular Expression* (Regex) untuk membantu proses pembersihan dan penyesuaian format data log agar lebih mudah dianalisis. Menurut Bintang et al., (2023), *Regular Expression* merupakan algoritma pemrograman yang digunakan untuk pengolahan teks seperti mencari, memanipulasi, serta memvalidasi pola tertentu di dalam suatu teks. Algoritma ini bekerja dengan cara mencocokkan pola yang telah ditentukan terhadap teks masukan secara iteratif, karakter demi karakter, hingga menemukan seluruh kecocokan yang ada. Penelitian Meiliyani et al., (2025) menunjukkan bahwa metode Regex terbukti dapat meningkatkan efisiensi dan akurasi pencarian teks dengan pola tertentu. Penelitian tersebut memanfaatkan metode Regex untuk merancang fitur pencarian produk pada aplikasi *e-commerce*. Penelitian Supartini (2023) juga menerapkan metode regex untuk mendeteksi serangan *SQL injection* pada website. Hasil dari penelitian tersebut menunjukkan peningkatan kecepatan dalam proses pencarian pada database serta meningkatnya ketepatan informasi yang berhasil diperoleh. Selain itu, penelitian Cahyo et al., (2025) menggunakan Regex untuk mengidentifikasi kata-kata yang berindikasi sebagai promosi judi online. Meskipun demikian, sebagian besar penelitian tersebut umumnya menitikberatkan pada pengembangan sistem *Security Information and Event Management* (SIEM) maupun *Intrusion Detection System* (IDS) untuk mendukung analisis keamanan jaringan. Proses prapengolahan (*preprocessing*) data log masih relatif sedikit dibahas. Padahal, kualitas data log pada tahap awal sangat memengaruhi efektivitas proses analisis serta monitoring keamanan jaringan.

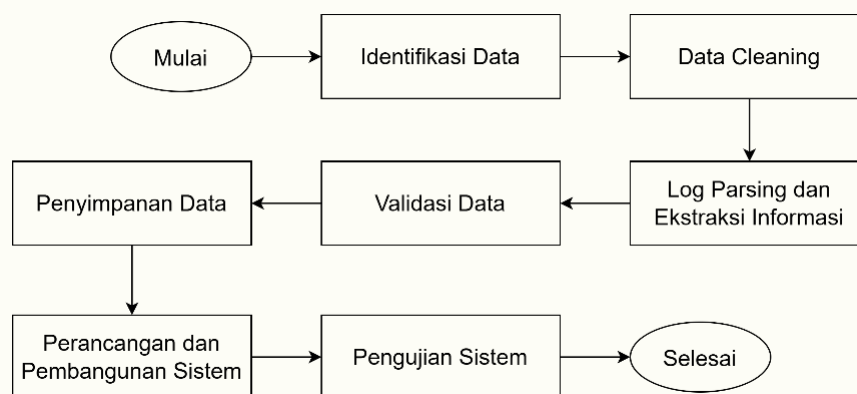
Proses parsing juga diperlukan untuk mengubah data *Syslog* yang semula tidak terstruktur menjadi informasi yang lebih mudah dianalisis. Zhang et al. (2023) menunjukkan bahwa proses *Syslog parsing* memiliki peran penting dalam mendukung analisis, diagnosis, dan pemantauan perangkat

jaringan. Selain pendekatan berbasis aturan seperti *Regex*, beberapa penelitian juga mengembangkan metode *log parsing* otomatis yang memanfaatkan *pattern recognition* dan analisis frekuensi untuk mengenali struktur log tanpa memerlukan pola yang didefinisikan secara manual (Sedki et al., 2021). Pendekatan *log parsing* pada penelitian ini juga sejalan dengan konsep yang digunakan dalam penelitian He et al. (2017) yang menjelaskan bahwa pesan log mentah umumnya perlu diparsing terlebih dahulu sebelum digunakan dalam proses analisis. Parsing dilakukan untuk mengubah pesan log menjadi struktur yang lebih terorganisasi sehingga informasi penting di dalamnya dapat diproses secara lebih efektif. Hal tersebut menunjukkan bahwa proses parsing merupakan tahap awal yang penting dalam pengolahan data log dalam jumlah besar.

Berdasarkan kondisi tersebut, dapat diketahui bahwa belum banyak penelitian yang melakukan eksplorasi terhadap pengelolaan data log secara menyeluruh dari tahap pembersihan hingga penyajian informasi. Oleh karena itu, penelitian ini bertujuan menganalisis penerapan *Regular Expression* sebagai metode *data cleaning* dan *log parsing* pada data *Syslog* untuk menghasilkan informasi yang lebih terstruktur. Penerapan tersebut mencakup normalisasi atribut *Tag*, *Time*, dan *Host*, ekstraksi informasi dari kolom *Message* berdasarkan pola *Common Event Format* (CEF), serta pengisian atribut yang kosong secara otomatis dengan memanfaatkan informasi yang tersedia dalam log. Selanjutnya, data hasil pengolahan diintegrasikan ke dalam basis data MySQL dan disajikan melalui antarmuka berbasis web untuk mendukung pencarian, penyaringan, dan pengelolaan data dalam jumlah besar. Dengan pendekatan tersebut, penelitian ini diharapkan dapat menghasilkan alur pengolahan *Syslog* yang terstruktur sekaligus mendukung kebutuhan monitoring keamanan jaringan.

METODE

Penelitian ini dilakukan melalui beberapa tahapan yang meliputi identifikasi data, *data cleaning*, *log parsing* dan ekstraksi informasi, validasi hasil pengolahan, penyimpanan data ke dalam *database*, serta perancangan dan pembangunan sistem berbasis web. Tahapan tersebut disusun secara berurutan untuk memastikan data *Syslog* yang semula memiliki format yang tidak konsisten dapat dibersihkan, dinormalisasi, dan diubah menjadi data yang lebih terstruktur sebelum digunakan dalam proses penyimpanan dan analisis. Alur tahapan penelitian secara keseluruhan ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Alur Penelitian

Identifikasi Data

Dataset yang digunakan dalam penelitian ini merupakan data log yang diperoleh dari lingkungan BRIN. Data dikumpulkan pada periode 4 Juni 2025 sampai 17 Juni 2025 dengan jumlah keseluruhan 6.534.086 record log. Data yang digunakan dalam penelitian ini dikelola oleh penulis pertama sehingga tidak memerlukan perizinan khusus dalam pemakaiannya di penelitian ini. Data berasal dari beberapa sumber perangkat yang mengirimkan log ke *Syslog server* BRIN dengan jenis perangkat disesuaikan dengan informasi yang tersedia. Data awal disimpan dalam format CSV dan memiliki beberapa atribut utama yaitu *IP*, *Time*, *Host*, *Facility*, *Priority*, *Tag*, dan *Message*. Atribut-atribut tersebut digunakan

sebagai dasar untuk mengidentifikasi karakteristik data serta menentukan bagian yang memerlukan proses normalisasi dan pembersihan.

Tahap identifikasi dilakukan dengan memeriksa struktur setiap atribut, pola nilai yang muncul, nilai kosong, serta ketidakkonsistenan format pada data. Berdasarkan hasil identifikasi, ditemukan beberapa permasalahan pada *dataset*, terutama pada atribut *Tag*, *Time*, dan *Host*. Atribut *Tag* dapat mengandung nilai yang tidak sesuai dengan format yang diharapkan, atribut *Time* memiliki format waktu yang tidak seragam, sedangkan atribut *Host* dapat berisi nilai yang tidak relevan atau kosong. Selain itu, informasi yang terdapat pada atribut *Message* digunakan sebagai sumber informasi tambahan untuk melengkapi atribut yang memiliki nilai kosong.

Data Cleaning

Tahap *data cleaning* dilakukan untuk meningkatkan konsistensi dan kualitas data sebelum proses analisis. Pada tahap ini, Regex digunakan untuk mengenali pola tertentu pada setiap atribut sehingga nilai yang tidak sesuai dapat diidentifikasi dan ditangani secara sistematis. Proses *data cleaning* diawali dengan proses pembacaan data dengan pemisah yang sesuai dengan struktur *dataset*. Nilai kolom yang secara tidak sengaja terbaca sebagai *Unnamed* diidentifikasi menggunakan Regex dan dikonversi menjadi nilai kosong (NaN). Setelah itu, nama atribut disesuaikan menjadi *IP*, *Time*, *Host*, *Facility*, *Priority*, *Tag*, dan *Message*.

Pembersihan atribut *Tag* dilakukan dengan mengidentifikasi nilai yang memiliki pola *timestamp*. Nilai yang teridentifikasi sebagai *timestamp* pada atribut *Tag* dianggap tidak sesuai dengan struktur atribut tersebut dan dikonversi menjadi nilai kosong. Pada atribut *Time*, dilakukan penyesuaian format waktu agar data yang ada menjadi lebih konsisten. Informasi tahun ditambahkan berdasarkan tahun sistem operasi apabila data waktu tidak mencantumkan tahun. Nilai yang tidak dapat dikonversi ke format waktu yang valid ditandai sebagai *Not a Time* (NaT). Sementara itu, pembersihan atribut *Host* dilakukan dengan mengidentifikasi nilai yang tidak sesuai, termasuk nama bulan dan *string noise*. Nilai *Host* yang kosong kemudian dilakukan imputasi menggunakan informasi *MAC Address* sebagai identitas alternatif yang tersedia pada atribut *Message*.

Log Parsing dan Ekstraksi Informasi

Setelah proses pembersihan atribut dasar selesai, dilakukan *log parsing* terhadap informasi pada atribut *Message*. Tahap ini bertujuan mengubah bagian log yang masih berbentuk teks menjadi informasi yang lebih terstruktur. Proses *parsing* secara khusus diterapkan pada data dengan alamat IP 192.168.100.148 yang memiliki pesan dengan struktur *Common Event Format* (CEF). Regex digunakan untuk memisahkan bagian-bagian CEF berdasarkan karakter pemisah (`|`). Hasil ekstraksi terdiri atas tujuh komponen utama, yaitu *CEF Version*, *Vendor*, *Product*, *Version*, *Process*, *Event*, dan *Severity*.

Informasi hasil *parsing* kemudian digunakan untuk memperkaya atribut *Host* dan *Tag*. Nama perangkat diekstraksi dari bagian awal *Event* menggunakan pola Regex dan digunakan untuk mengisi atribut *Host* apabila atribut tersebut sebelumnya kosong. Selain itu, informasi CEF berupa *CEF Version*, *Vendor*, *Product*, dan *Version* digunakan untuk membentuk nilai *Tag* pada data yang memenuhi kondisi yang telah ditentukan.

Pola Regular Expression

Pola Regex digunakan sesuai dengan karakteristik data yang ditemukan pada *dataset*. Setiap pola memiliki fungsi berbeda dalam proses *data cleaning* dan *log parsing*. Pola Regex yang digunakan dalam penelitian ini ditunjukkan pada Tabel 1. Pola tersebut diterapkan pada atribut yang sesuai dengan karakteristik data. Penggunaan Regex memungkinkan proses identifikasi pola dilakukan secara konsisten sehingga data yang memiliki format tidak sesuai dapat ditangani sebelum digunakan pada tahap berikutnya.

Tabel 1 Pola Regular Expression pada proses Data Cleaning dan Log Parsing

No	Proses	Pola Regex	Fungsi
1	Identifikasi <i>Unnamed</i>	<code>^Unnamed.*</code>	Mengidentifikasi nama kolom atau nilai yang diawali dengan <i>Unnamed</i> untuk dikonversi menjadi nilai kosong (NaN).
2	Pembersihan <i>Tag</i>	<code>(*\\w{3}\\s+)?\\d{1,2}\\s+\\d{2}:\\d{2}:\\d{2}</code>	Mengidentifikasi nilai dengan pola timestamp pada atribut <i>Tag</i> .
3	Pembersihan <i>Host</i>	<code>(?i)^*?(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec)\$</code>	Mengidentifikasi nama bulan yang tidak sesuai pada atribut <i>Host</i> .
4	Ekstraksi MAC Address	<code>([0-9a-fA-F]{4}\\.[0-9a-fA-F]{4})</code>	Mengekstraksi MAC Address dari atribut <i>Message</i> untuk mengisi atribut <i>Host</i> yang kosong.
5	Parsing CEF	<code>CEF:([\\^]+)\\ ([\\^]+)\\ ([\\^]+)\\ ([\\^]+)\\ ([\\^]+)\\ ([\\^]+)\\ ([\\^]+)\\ ([\\^]+)</code>	Memisahkan isi <i>Message</i> yang memiliki format CEF berdasarkan <i>delimiter</i> () untuk mengekstraksi tujuh komponen informasi yaitu CEF Version, Vendor, Product, Version, Process, Event, dan Severity.
6	Ekstraksi Perangkat	<code>^([\\^\\s]+)\\s+</code>	Mengambil nama perangkat yang berada pada bagian awal nilai <i>Event</i> . Informasi tersebut digunakan untuk melengkapi atribut <i>Host</i> apabila nilai <i>Host</i> sebelumnya kosong.

Validasi Hasil Data Cleaning dan Log Parsing

Setelah proses *data cleaning* dan *log parsing*, dilakukan pemeriksaan terhadap hasil pengolahan untuk memastikan bahwa transformasi data berjalan sesuai dengan aturan yang telah ditentukan. Validasi dilakukan dengan memeriksa struktur atribut, keberadaan nilai kosong, kesesuaian format waktu, serta hasil ekstraksi informasi dari atribut *Message*. Validasi juga dilakukan dengan membandingkan kondisi data sebelum dan sesudah proses pembersihan. Pemeriksaan difokuskan pada atribut *Tag*, *Time*, dan *Host*, serta data yang diproses menggunakan pola CEF. Hasil validasi digunakan untuk memastikan bahwa nilai yang berhasil diekstraksi dapat ditempatkan pada atribut yang sesuai dan tidak mengubah struktur atribut lain yang tidak menjadi target proses pembersihan. Data hasil pengolahan kemudian disimpan ke dalam format CSV menggunakan pemisah titik koma (*semicolon*) agar dapat disimpan pada basis data.

Penyimpanan Data ke Database

Data yang telah melalui proses *data cleaning* dan *log parsing* selanjutnya disimpan ke dalam MySQL sebagai sistem manajemen basis data. Penyimpanan ke dalam basis data dilakukan untuk memudahkan pengelolaan, pencarian, serta pengambilan data pada sistem berbasis web. Pengelolaan basis data dilakukan menggunakan phpMyAdmin melalui lingkungan XAMPP.

Perancangan dan Pembangunan Sistem

Pada tahap ini, sistem dirancang untuk dapat menampilkan, mengelola, serta memantau data log server secara terpusat melalui antarmuka berbasis web. Sistem ini akan memiliki 2 halaman utama yaitu halaman *dashboard* yang dilengkapi dengan fitur pencarian dan filter, serta halaman *upload file*. Arsitektur sistem terdiri atas bagian antarmuka yang dibangun menggunakan PHP, CSS dan JavaScript, serta bagian basis data yang dikelola menggunakan MySQL melalui lingkungan pengembangan XAMPP. Pemilihan XAMPP sebagai web server didasarkan pada kemampuannya dalam menyediakan proses instalasi dan konfigurasi yang sederhana sehingga mempermudah pengembang dalam membangun dan menguji sistem. MySQL telah memantapkan posisinya sebagai salah satu basis data yang paling populer dan banyak digunakan di industri, serta dikenal karena kecepatan dan

keandalannya (Salunke & Ouda, 2024). Selain itu, XAMPP bersifat *open source* dan terintegrasi dengan berbagai alat bantu seperti phpMyAdmin yang mendukung pengelolaan *database* secara efektif dan efisien (Teneke et al., 2024).

Alur kerja sistem dimulai dari proses memasukkan data log server yang telah melalui tahap pembersihan data ke dalam *database* MySQL. Data yang tersimpan kemudian diproses dan ditampilkan kembali melalui antarmuka pengguna dalam bentuk *dashboard* interaktif sehingga pengguna dapat melihat, mencari, serta memfilter data log server sesuai kebutuhan. Antarmuka pengguna dirancang dengan memanfaatkan DataTables. Penggunaan DataTables memberikan berbagai manfaat, salah satunya memiliki fitur seperti navigasi antar halaman. Ketika jumlah data dalam tabel melebihi batas tertentu, data akan dibagi ke halaman berikutnya, dan pengguna dapat berpindah halaman dengan menekan tombol navigasi. Selain itu, terdapat fitur pencarian cepat yang memudahkan pengguna menemukan data tertentu ketika jumlah data sangat banyak. DataTables juga telah mendukung tampilan responsif sehingga tabel dapat menyesuaikan ukuran layar saat web diakses melalui berbagai perangkat (Ramadhan & Putro, 2020).

Pengujian Sistem

Pengujian dilakukan untuk memastikan bahwa seluruh fungsi berjalan dengan baik dan memenuhi spesifikasi yang ditetapkan (Saifudin et al., 2022). Setiap menu yang telah dikembangkan diuji untuk menilai kinerja dan keandalannya. Proses ini mencakup pengujian tampilan *dashboard* yang menampilkan data log server dalam bentuk tabel, fitur pencarian (*search*), filter data berdasarkan rentang waktu tertentu, serta halaman unggah (*upload*) untuk memasukkan *file* yang telah melalui proses pembersihan data ke *database*. Pengujian dilakukan dengan metode *Black Box Testing* untuk menilai apakah setiap fitur berfungsi sesuai dengan yang diharapkan. Menurut Maspupah (2022), *Black Box Testing* berfokus pada fungsionalitas sistem yang mencakup pengujian terhadap skenario langka dan kondisi *error*. *Black Box Testing* juga dapat membantu mengidentifikasi potensi masalah atau kesalahan (*bug*) yang mungkin terjadi pada aplikasi. Melalui proses pengujian yang dilakukan secara menyeluruh, risiko terjadinya kesalahan atau kegagalan dalam penggunaan aplikasi (Salih & Saefullah, 2024). Pendekatan ini bertujuan untuk memastikan bahwa sistem beroperasi sesuai tujuan tanpa memerlukan pemahaman terhadap struktur kode program.

HASIL DAN PEMBAHASAN

Hasil Evaluasi Kode Pembersihan Data

1. Load Data

Proses pembacaan data dilakukan dengan menggunakan pemisah tab dan menunjukkan hasil yang optimal di mana setiap kolom berhasil teridentifikasi dan terpisah sesuai dengan format yang ditentukan. Hal ini mengindikasikan bahwa struktur dataset telah dikenali secara tepat oleh sistem sehingga data siap digunakan untuk proses pengolahan selanjutnya.

2. Konversi Nilai 'Unnamed' Menjadi NaN

Ditemukan data yang mengandung nilai 'Unnamed' akibat proses load data yang dilakukan tanpa menetapkan *header* yang sesuai. Kondisi ini menyebabkan sistem secara otomatis menjadikan baris pertama pada data sebagai nama kolom sebelum parameter *header* diatur ke nilai 'none'. Data yang semula memiliki atribut dengan nilai kosong secara otomatis terdeteksi sebagai 'Unnamed'. Oleh karena itu, diperlukan penanganan terhadap nilai yang tidak sesuai tersebut dengan mengganti seluruh nilai yang diawali kata 'Unnamed' pada DataFrame menjadi NaN (*Not a Number*) menggunakan pendekatan Regex.

3. Penyesuaian Nama Kolom

Data sudah berhasil diurutkan sesuai dengan format data yang dihasilkan dari *Visual Syslog Server*. Atribut pada data tersebut diurutkan menjadi *IP*, *Time*, *Host*, *Facility*, *Priority*, *Tag*, dan *Message* (Gambar 2). Penyesuaian ini memastikan konsistensi struktur data dengan format sistem *Syslog server* sehingga memudahkan proses analisis dan interpretasi pada tahap berikutnya.

IP	Time	Host	Facility	Priority	Tag	Message
192.168.100.148	Jun 04 11:14:23	Jun	NaN	NaN	NaN	04 11:14:26.669 WIB 192.168.100.148 CEF:1 Ubiq...
10.33.0.20	Jun 4 11:14:37	2023RUMPINGDDSIGNCENTERLT101	daemon	info	602232f43dc9,U6-LR-6.7.17+15512	logread[7901]: Logread connected to 10.34.0.2:514
10.33.0.28	Jun 4 11:14:38	2023RUMPINGDDISEMINASILT101	daemon	info	602232f3294d,U6-LR-6.7.17+15512	logread[22709]: Logread connected to 10.34.0.2...
10.33.0.20	Jun 4 11:14:40	2023RUMPINGDDSIGNCENTERLT101	daemon	err	602232f43dc9,U6-LR-6.7.17+15512	procd: Process didn't stop on SIGTERM - sendin...
10.33.0.20	Jun 4 11:14:40	2023RUMPINGDDSIGNCENTERLT101	daemon	err	602232f43dc9,U6-LR-6.7.17+15512	mcad: mcad[6178]: inform-prepare.dump_radio_st...

Gambar 2. DataFrame dengan Nama Atribut yang Telah Sesuai

4. Pembersihan Atribut Tag

Data pada kolom *Tag* menunjukkan adanya variasi format dan nilai yang tidak konsisten, seperti kemunculan karakter non-alfanumerik, *string noise*, serta data kosong. Untuk mengatasi permasalahan tersebut, dilakukan proses pembersihan data dengan memanfaatkan metode regex. Dalam proses ini, regex digunakan untuk menyalin data dan membersihkan kolom *Tag* dari nilai dengan format waktu (*timestamp*) yang kemudian menggantinya menjadi NaN agar lebih mudah diidentifikasi. Pola regex yang digunakan mampu mengenali format penulisan timestamp seperti "Jan 4 11:32:51" yang umumnya muncul pada log sistem. Setelah proses pembersihan diterapkan, data pada kolom *Tag* menjadi lebih konsisten dan siap digunakan untuk analisis selanjutnya.

5. Pembersihan Atribut Time

Kolom *Time* sering kali menyimpan data *timestamp* dengan format tidak konsisten seperti penulisan tanggal satuan yang bervariasi dan tidak mencantumkan informasi tahun (Gambar 3). Untuk menjaga konsistensi, dilakukan proses normalisasi waktu dengan menambahkan informasi tahun secara otomatis berdasarkan tahun sistem operasi serta menyesuaikan format agar mengikuti standar *datetime* yaitu YYYY-MM-DD HH:MM:SS. Selain itu, nilai waktu yang tidak dapat dinormalisasi ditandai sebagai NaT (*Not a Time*) untuk memudahkan proses penyaringan dan penanganan data tidak valid. Setelah tahap pembersihan selesai, seluruh data pada atribut *Time* berhasil dikonversi ke dalam tipe data *datetime* yang valid seperti yang ditunjukkan pada Gambar 4.

Time	IP	Host	Facility	Priority	Tag	Message
Jun 4 11:31:35	10.33.0.14	2023RUMPINGDLABDOLT101	kern	warning	602232f43331,U6-LR-6.7.17+15512	kernel: [3311685.643945] addbarsp,(our)txba=64...
Jun 04 11:43:13	10.0.0.22	*Jun	local7	warning	NaN	%OPTICAL_MODULE-4: ALARM: Optical module is abn...
Jun 4 11:35:34	10.33.0.20	2023RUMPINGDDSIGNCENTERLT101	kern	warning	602232f43dc9,U6-LR-6.7.17+15512	kernel: [3397603.791944] a4, flush one!
Jun 04 11:32:40	10.10.170.254	NaN	local7	info	NaN	%APMG-6-STA_CHANGE: Client(ca6e.ad61.0db3) mov...
Jun 4 11:45:56	10.33.0.19	2023RUMPINGDAVIONIKT202	kern	warning	602232f43e11,U6-LR-6.7.17+15512	kernel: [241717.887219] 80211> CFG80211_OpeSta...

Gambar 3. Format Tidak Konsisten pada Kolom Time

Time	IP	Host	Facility	Priority	Tag	Message
2025-06-04 11:14:23	192.168.100.148	Jun	NaN	NaN	NaN	04 11:14:26.669 WIB 192.168.100.148 CEF:1 Ubiq...
2025-06-04 11:14:37	10.33.0.20	2023RUMPINGDDSIGNCENTERLT101	daemon	info	602232f43dc9,U6-LR-6.7.17+15512	logread[7901]: Logread connected to 10.34.0.2:514
2025-06-04 11:14:38	10.33.0.28	2023RUMPINGDDISEMINASILT101	daemon	info	602232f3294d,U6-LR-6.7.17+15512	logread[22709]: Logread connected to 10.34.0.2...
2025-06-04 11:14:40	10.33.0.20	2023RUMPINGDDSIGNCENTERLT101	daemon	err	602232f43dc9,U6-LR-6.7.17+15512	procd: Process didn't stop on SIGTERM - sendin...
2025-06-04 11:14:40	10.33.0.20	2023RUMPINGDDSIGNCENTERLT101	daemon	err	602232f43dc9,U6-LR-6.7.17+15512	mcad: mcad[6178]: inform-prepare.dump_radio_st...

Gambar 4. Hasil Formatting Datetime pada Kolom Time

6. Pembersihan Atribut Host

Kolom *Host* pada data Syslog kerap berisi nilai yang tidak relevan seperti nama bulan, *string noise*, atau bahkan nilai kosong (Gambar 5). Sebagai solusi, pembersihan atribut *Host* dilakukan dengan regex untuk menghapus nilai yang mengandung *noise* seperti nama bulan atau karakter acak yang tidak

sesuai format *Host*. Pada baris data dengan nilai *Host* kosong, dilakukan proses ekstraksi informasi dari kolom *Message* seperti *device name* atau *MAC address* untuk mengisi kolom *Host* secara otomatis. Langkah ini akan menghilangkan nilai kosong pada kolom *Host* secara signifikan.

Time	IP	Host	Facility	Priority	Tag	Message
2025-06-04 11:14:23	192.168.100.148		Jun	NaN	NaN	04 11:14:26.669 WIB 192.168.100.148 CEF:1 Ubiqu...

Gambar 5. Atribut Host yang Masih terdapat Nama Bulan

Setelah menelusuri atribut *Message* lebih lanjut, ternyata terdapat beberapa informasi penting. Salah satunya informasi terkait *MAC Address* dengan format XXXX.XXXX.XXXX. Informasi tersebut kemudian diekstraksi dan digunakan untuk mengisi nilai pada kolom *Host* yang sebelumnya kosong sehingga dapat melengkapi data secara otomatis dan meningkatkan konsistensi atribut. Hasil ekstraksi tersebut dapat dilihat pada Gambar 6.

Time	IP	Host	Facility	Priority	Tag	Message
2025-06-04 11:30:40	10.10.170.254	1efe.1c6b.325d	local7	info	NaN	%APMG-6-STA_CHANGE: Client(1efe.1c6b.325d) mov...
2025-06-04 11:30:41	10.10.170.254	eec0.c87e.c66e	local7	info	NaN	%ROAMING-6-ROAM_EVENT: Client(eec0.c87e.c66e) ...
2025-06-04 11:30:42	10.10.170.254	fe7b.24f8.cc28	local7	info	NaN	%APMG-6-STA_CHANGE: Client(fe7b.24f8.cc28) mov...
2025-06-04 11:30:42	10.10.170.254	c6d5.4420.bb9e	local7	info	NaN	%APMG-6-STA_DEL: Client(c6d5.4420.bb9e) notify...
2025-06-04 11:30:42	10.10.170.254	6e38.9869.14a6	local7	info	NaN	%ROAMING-6-ROAM_EVENT: Client(6e38.9869.14a6) ...

Gambar 6. Pengisian Kolom Host dengan Informasi MAC Address

7. Ekstrak Informasi Penting dari Kolom Message

Pada tahap ini, dilakukan proses ekstraksi informasi penting dari atribut *Message* khusus untuk data dengan alamat IP 192.168.100.148. Proses ini bertujuan untuk memperkaya data dengan menambahkan informasi yang lebih terstruktur ke dalam kolom *Tag* dan *Host*, serta menyusun ulang isi *Message* agar lebih ringkas dan mudah dipahami. Proses ekstraksi informasi dari log ini merupakan bagian dari *log parsing*, yaitu proses mengubah data log mentah menjadi format yang lebih terstruktur sehingga dapat dianalisis secara lebih efisien dan akurat (Huang et al., 2025). Langkah pertama dimulai dengan membuat mask untuk memilih baris data dengan IP tertentu yaitu 192.168.100.148. Setelah itu, isi atribut *Message* pada baris tersebut dipisahkan berdasarkan struktur *Common Event Format* (CEF) menggunakan pola regex. Pola ini dirancang untuk membagi log berdasarkan karakter pemisah digunakan dalam format CEF sehingga setiap bagian log dapat diidentifikasi dengan jelas. Hasil pemisahan ini menghasilkan tujuh komponen utama, yaitu *CEF_Version*, *Vendor*, *Product*, *Version*, *Process*, *Event*, dan *Severity*. Selanjutnya, hasil ekstraksi tersebut dimanfaatkan untuk memperbarui kolom lain. Didefinisikan pola regex yang akan mengekstraksi informasi *device* dari bagian awal teks *Event* pada *Message*. Kolom *Host* diisi dengan nama perangkat tersebut hanya jika sebelumnya belum memiliki nilai (Gambar 7). Setelah itu, kolom *Tag* diisi dengan kombinasi informasi CEF untuk baris yang memenuhi kondisi IP dengan nilai 192.168.100.148 serta kolom *Tag* dengan nilai NaN (Gambar 8).

Time	IP	Host	Facility	Priority	Tag	Message
2025-06-04 11:14:23	192.168.100.148		NaN	NaN	CEF:1 Ubiquiti UniFi Network 9.0.114	admin_activity-683fc822c878422526d45bdb riki...

Gambar 7. Kolom Host Diisi oleh Nama Perangkat

Time	IP	Host	Facility	Priority	Tag	Message
2025-06-04 11:14:23	192.168.100.148	riki007	NaN	NaN	CEF:1 Ubiquiti UniFi Network 9.0.114	admin_activity-683fc822c878422526d45bdb riki...
2025-06-04 11:14:57	192.168.100.148	RedmiS2-Redmi	NaN	NaN	CEF:1 Ubiquiti UniFi Network 9.0.114	client-683fc844c878422526d45d72 RedmiS2-Redm...
2025-06-04 11:15:22	192.168.100.148	V2250	NaN	NaN	CEF:1 Ubiquiti UniFi Network 9.0.114	client-683fc85dc878422526d46e2e V2250 roamed...
2025-06-04 11:15:26	192.168.100.148	MSI	NaN	NaN	CEF:1 Ubiquiti UniFi Network 9.0.114	client-683fc860c878422526d46e49 MSI disconne...
2025-06-04 11:15:29	192.168.100.148	MSI	NaN	NaN	CEF:1 Ubiquiti UniFi Network 9.0.114	client-683fc863c878422526d46e61 MSI connecte...

Gambar 8. Kolom Tag Diisi dengan Kombinasi Informasi CEF

8. Penyimpanan Data Setelah Proses Pembersihan

Tahap ini menyalin data hasil pembersihan ke 659omputer baru. Selanjutnya, seluruh nilai kosong (NaN) digantikan dengan 'NULL' agar dapat dikenali sebagai nilai kosong oleh sistem *database*. Kemudian data hasil pembersihan disimpan ke dalam berkas dengan nama *dataclean* dalam format CSV. Format CSV banyak digunakan sebagai format pertukaran data yang umum di sistem basis data dan aplikasi analisis karena didukung secara luas oleh berbagai sistem DBMS serta perangkat lunak lain sehingga memudahkan integrasi dan interoperabilitas data (Liu et al., 2025). Berkas tersebut disimpan tanpa menyertakan *header* dan indeks, serta menggunakan tanda titik koma (*semicolon*) sebagai pemisah antar kolom.

Evaluasi Hasil *Data Cleaning* dan *Log Parsing*

Evaluasi dilakukan untuk mengetahui hasil penerapan *Regular Expression* (Regex) pada proses *data cleaning* dan *log parsing*. Evaluasi dilakukan dengan membandingkan kondisi data sebelum dan sesudah proses pengolahan pada dataset yang terdiri atas 6.534.086 record. Pada atribut *Tag*, evaluasi dilakukan dengan menghitung jumlah data yang memiliki pola *timestamp* sebelum proses pembersihan dan jumlah data yang berhasil diidentifikasi serta dikonversi menjadi nilai kosong setelah penerapan Regex. Hasil tersebut digunakan untuk mengetahui kemampuan pola Regex dalam mengidentifikasi nilai yang tidak sesuai dengan karakteristik atribut *Tag*. Pada atribut *Time*, evaluasi dilakukan dengan menghitung jumlah data yang berhasil dinormalisasi ke dalam format *datetime* serta jumlah data yang tidak dapat dikonversi dan ditandai sebagai *NaT*. Hasil normalisasi menunjukkan tingkat keberhasilan proses penyesuaian format waktu sehingga data dapat digunakan untuk proses pencarian dan penyaringan berdasarkan rentang waktu.

Sementara itu, evaluasi pada atribut *Host* dilakukan terhadap nilai yang tidak sesuai, nilai kosong, serta informasi yang berhasil dipulihkan dari atribut *Message*. Regex digunakan untuk mengidentifikasi nama bulan yang terdapat pada atribut *Host*, sedangkan pola *MAC Address* digunakan untuk mengekstraksi informasi dari *Message*. Hasil ekstraksi tersebut kemudian digunakan untuk melengkapi atribut *Host* yang sebelumnya kosong. Pada proses *log parsing*, evaluasi dilakukan dengan menghitung jumlah *Message* yang memiliki struktur CEF dan jumlah data yang berhasil dipisahkan menjadi tujuh komponen yaitu *CEF Version*, *Vendor*, *Product*, *Version*, *Process*, *Event*, dan *Severity*. Hasil tersebut digunakan untuk mengetahui keberhasilan proses ekstraksi informasi dari log yang memiliki format CEF.

Selain itu, hasil ekstraksi nama perangkat dari bagian *Event* juga dievaluasi berdasarkan jumlah data yang berhasil memperoleh informasi *device*. Informasi tersebut digunakan untuk melengkapi atribut *Host* apabila nilai *Host* sebelumnya kosong. Secara keseluruhan, evaluasi tersebut menunjukkan bahwa Regex dapat digunakan untuk melakukan identifikasi pola, pembersihan nilai yang tidak sesuai, normalisasi, serta ekstraksi informasi dari data *Syslog*. Pengukuran berdasarkan jumlah data sebelum dan sesudah pengolahan pada Tabel 2 memberikan 659omputer yang lebih objektif mengenai hasil penerapan Regex dibandingkan hanya berdasarkan tampilan data atau contoh hasil pengolahan.

Tabel 2 Jumlah Record Data dalam Setiap Proses *Data Cleaning* dan *Log Parsing*

No	Proses Evaluasi	Hasil Evaluasi	Jumlah Record
1	Data awal	Total dataset	6.534.086
2	Pembersihan <i>Tag</i>	<i>Tag</i> terdeteksi memiliki pola <i>timestamp</i>	1.474.104
3	Pembersihan <i>Tag</i>	<i>Tag</i> berhasil dikonversi menjadi <i>NaN</i>	1.474.104
4	Normalisasi <i>Time</i>	Berhasil dikonversi ke format <i>datetime</i>	6.534.086
5	Normalisasi <i>Time</i>	Gagal dikonversi / <i>NaT</i>	0
6	Pembersihan <i>Host</i>	<i>Host</i> terdeteksi sebagai nama bulan	601.017

No	Proses Evaluasi	Hasil Evaluasi	Jumlah Record
7	Pembersihan <i>Host</i>	<i>Host</i> berhasil dibersihkan	601.017
8	Ekstraksi MAC Address <i>Message</i> mengandung pola MAC Address		1.133.792
9	Pelengkapan <i>Host</i>	<i>Host</i> dilengkapi menggunakan MAC Address	1.133.792
10	<i>Log parsing</i> CEF	Record dengan IP 192.168.100.148	39.396
11	<i>Log parsing</i> CEF	<i>Message</i> berhasil diparsing	39.396
12	Ekstraksi <i>Device</i>	<i>Device Name</i> berhasil diekstraksi	39.396
13	Pelengkapan <i>Host</i>	<i>Host</i> dilengkapi menggunakan <i>Device Name</i>	39.396

Penyimpanan Data ke DBMS

Setelah proses pembersihan dilakukan pada log server, data yang telah dipastikan bersih dan terstruktur kemudian disiapkan untuk tahap penyimpanan ke sistem basis data (DBMS). *Database* berperan penting dalam mendukung proses pengambilan 660omputer660, tidak hanya sebagai media penyimpanan, tetapi juga dalam mengolah serta menyajikan informasi yang relevan bagi pengambil kebijakan melalui proses pengolahan dan penyaringan data yang terstruktur (Aulia et al., 2023).

Pada penelitian ini, *Database Management System* (DBMS) yang digunakan 660omput MySQL yang dikelola melalui platform phpMyAdmin. Data disimpan dengan format dan struktur yang telah disesuaikan, serta menggunakan pemisah titik koma agar sesuai dengan standar impor data ke dalam sistem basis data. Adapun 660ompute-langkah penerapan pembuatan dan penyimpanan ke *database* dijelaskan sebagai berikut.

1. Buka phpMyAdmin melalui XAMPP dengan mengetikkan 660omput <http://localhost/660omputer660n> pada *browser*.
2. Masukkan nama *database* yang diinginkan pada menu *Database*, kemudian klik *Create*.
3. Setelah itu, buat tabel dengan memilih nama *database* yang baru saja dibuat.
4. Masuk ke tabel yang sudah dibuat, lalu klik *impor* untuk memasukkan data yang sudah dibersihkan.
5. Data yang berhasil dimasukkan akan ditampilkan pada halaman utama *database* seperti pada Gambar 9.

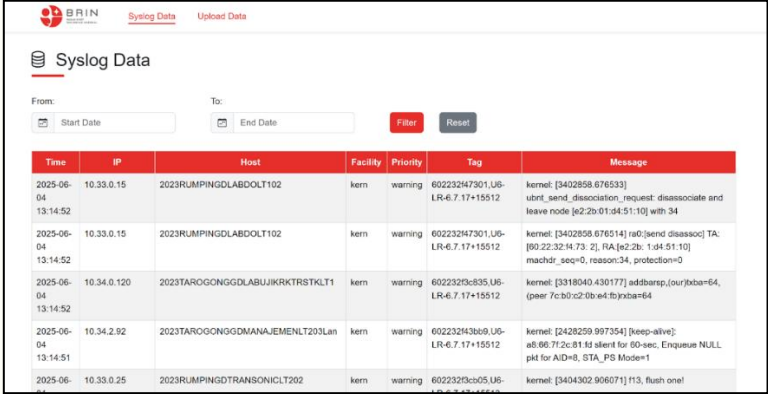
Time	IP	Host	Facility	Priority	Tag	Message
2025-06-04 11:14:23	192.168.130.148	rs007	NETL	INFO	CEF-13:logd[100]	admin_activity:603a822e7842550645db (rs007)
2025-06-04 11:14:37	10.33.9.20	2023RUMPMINGDISIGNCENTERLT101	daemon	info	logread[7907]	Logread connected to 10.34.0.2514
2025-06-04 11:14:38	10.33.9.20	2023RUMPMINGDISIGNCENTERLT101	daemon	info	logread[22700]	Logread connected to 10.34.0.2514
2025-06-04 11:14:40	10.33.9.20	2023RUMPMINGDISIGNCENTERLT101	daemon	err		procd: Process didn't stop on SIGTERM - sending SI...
2025-06-04 11:14:40	10.33.9.20	2023RUMPMINGDISIGNCENTERLT101	daemon	err		mcast: mcast[178] Inform...
2025-06-04 11:14:41	10.33.9.20	2023RUMPMINGDISIGNCENTERLT101	daemon	err		procd: Process didn't stop on SIGTERM - sending SI...
2025-06-04 11:14:43	10.33.9.20	2023RUMPMINGDISIGNCENTERLT101	kernel	warning		kernel: [3396302] 980702: nat@new_auth_req TA...
2025-06-04 11:14:43	10.33.9.20	2023RUMPMINGDISIGNCENTERLT101	kernel	warning		kernel: [3396302] 981103: nat@new_auth_req TA...

Gambar 9. Tampilan Data yang Sudah Dimasukkan ke DBMS

Pembangunan Interface Sistem

Antarmuka pengguna (*User Interface*) merupakan bagian terpenting dari sebuah sistem komputer karena berfungsi sebagai jembatan yang memungkinkan pengguna berinteraksi dan berkomunikasi dengan komputer. Pengalaman *User Interface* (UI) sangat berpengaruh dalam menunjang kepuasan dan efisiensi pengguna pada aplikasi analisis data. Dengan menggunakan *System Usability Scale*, UI yang baik terbukti meningkatkan penerimaan pengguna dan memudahkan proses analisis data secara

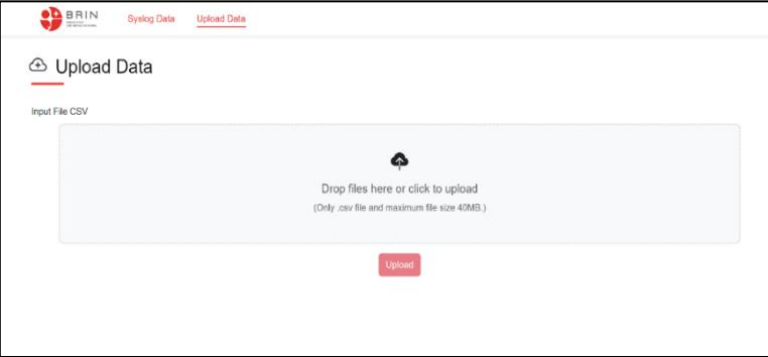
signifikan (Junaedi et al., 2023). Pada penelitian ini, dikembangkan antarmuka berbasis web yang memudahkan pengguna dalam melakukan analisis data Syslog hasil pembersihan. Tampilan awal *website* berupa *Dashboard* yang menyediakan ringkasan data serta menu navigasi ke fitur-fitur utama dapat dilihat pada Gambar 10.



Time	IP	Host	Facility	Priority	Tag	Message
2025-06-04 13:14:52	10.33.0.15	2023RUMPINGDLABDOLT102	kern	warning	60223247301;U6-LR-6.7.17+15512	kernel: [3402858.676533] ubrt_send_dissociation_request: dissociate and leave node [e2:2b:01:d4:51:10] with 34
2025-06-04 13:14:52	10.33.0.15	2023RUMPINGDLABDOLT102	kern	warning	60223247301;U6-LR-6.7.17+15512	kernel: [3402858.676514] ra0:[send dissociat] TA: [0:22:32:14:73:2], RA:[e2:2b:01:d4:51:10] machdr_seq=0, reason=34, protection=0
2025-06-04 13:14:52	10.34.0.120	2023TAROGONGDLABUJIKRTRSTKLT1	kern	warning	60223235835;U6-LR-6.7.17+15512	kernel: [3318040.430177] addbasp_our[jaba=64, peer 7c:b0:c2:0b:e4:fyxban=64
2025-06-04 13:14:51	10.34.2.92	2023TAROGONGODMANAJEMENT03Lan	kern	warning	60223243868;U6-LR-6.7.17+15512	kernel: [2428259.997354] [keep-alive]: a6:66:71:2c:81:fd silent for 60-sec, Enqueue NULL pkt for AID=8, STA_PS Mode=1
2025-06-04 13:14:51	10.33.0.25	2023RUMPINGDTRANSNCLT202	kern	warning	60223235835;U6-LR-6.7.17+15512	kernel: [3404302.906071] f13, flush one!

Gambar 10. Halaman *Dashboard Website Analisis Data Syslog*

Selain halaman *Dashboard*, terdapat juga fitur unggah data yang memungkinkan pengguna mengimpor file data langsung ke DBMS melalui halaman khusus (Gambar 11). Fitur ini dirancang untuk mempercepat proses integrasi data ke dalam sistem.



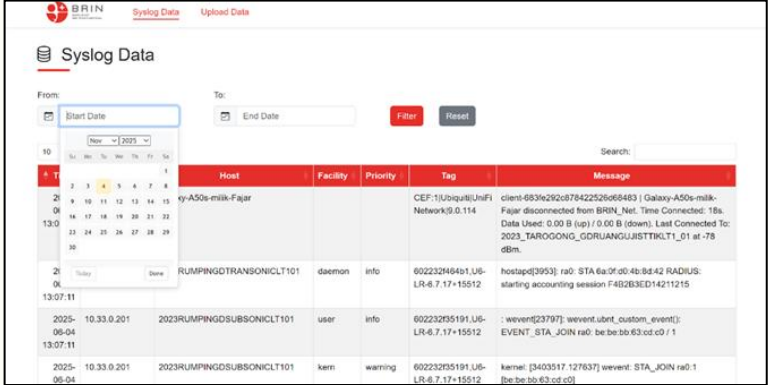
Input File CSV

Drop files here or click to upload
(Only .csv file and maximum file size 40MB.)

Upload

Gambar 11. Halaman *Upload Data Website Analisis Data Syslog*

Setelah data diunggah, tersedia halaman filter data yang memungkinkan pengguna mencari atau memfilter berdasarkan rentang waktu yang diinginkan. Visualisasi terkait proses filter dapat dilihat pada Gambar 12.



Time	IP	Host	Facility	Priority	Tag	Message
2025-06-04 13:07:11	10.33.0.201	2023RUMPINGDTRANSNCLT101	daemon	info	60223246481;U6-LR-6.7.17+15512	client-653a262c878422526c68483 Galaxy-A50s-milk-Fajar disconnected from BRIN_Net. Time Connected: 18s. Data Used: 0.00 B (up) / 0.00 B (down). Last Connected To: 2023_TAROGONG_ODRUANGUJISTTKLT1_01 at -78 dBm.
2025-06-04 13:07:11	10.33.0.201	2023RUMPINGDTRANSNCLT101	daemon	info	60223246481;U6-LR-6.7.17+15512	hostapd[3953]: ra0: STA 6a:0f:d0:4b:8d:42 RADIUS: starting accounting session F482B3ED14211215
2025-06-04 13:07:11	10.33.0.201	2023RUMPINGDTRANSNCLT101	user	info	60223295191;U6-LR-6.7.17+15512	: wevent[23707]: wevent.ubrt_custom_event(): EVENT_STA_JOIN ra0: be:be:bb:63:c0:c0 / 1
2025-06-04 13:07:11	10.33.0.201	2023RUMPINGDTRANSNCLT101	kern	warning	60223295191;U6-LR-6.7.17+15512	kernel: [3403517.127637] wevent: STA_JOIN ra0:1 (be:be:bb:63:c0:c0)

Gambar 12. Fitur Filter Data *Website Analisis Data Syslog*

Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur utama berjalan sesuai dengan fungsinya. Hasil pengujian terhadap lima skenario uji (*test case*) yang mencakup proses unggah file, validasi format, pencarian data, dan penyaringan berdasarkan waktu dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengujian *Black Box Testing*

ID	Test Case	Hasil yang Diharapkan	Hasil Uji	Kesimpulan
TC-01	Mengunggah file log dengan format CSV valid yang sudah melalui proses	File berhasil diunggah, data tersimpan ke <i>database</i> dan tampil di <i>dashboard</i>	File berhasil diunggah ke <i>database</i> dan tampil di <i>dashboard</i>	Berhasil
TC-02	Mengunggah file dengan format selain CSV (misalnya .txt atau .xls)	Sistem menolak unggahan dan menampilkan pesan eror "Format file tidak didukung"	Sistem menolak unggahan dan menampilkan pesan eror yang sesuai	Berhasil
TC-03	Mencari data pada kolom <i>Facility</i> dengan kata kunci " <i>daemon</i> "	Tabel menampilkan hanya baris dengan nilai <i>Facility</i> = " <i>daemon</i> "	Tabel menampilkan hanya baris dengan nilai <i>Facility</i> = " <i>daemon</i> "	Berhasil
TC-04	Mengatur filter waktu dengan format valid pada rentang waktu tertentu (misalnya 2025-10-01 sampai 2025-10-31)	Data log yang muncul sesuai rentang waktu yang dimasukkan	Data log yang muncul sesuai rentang waktu yang dimasukkan	Berhasil
TC-05	Melakukan pencarian data dengan kata kunci yang tidak ada dalam <i>database</i>	Sistem menampilkan pesan "Data tidak ditemukan"	Sistem berhasil menampilkan pesan yang sesuai	Berhasil

Pembahasan

Hasil penelitian menunjukkan bahwa penerapan *Regular Expression* (Regex) dapat mendukung proses *data cleaning* dan *log parsing* pada data *Syslog*. Pada proses *data cleaning*, Regex digunakan untuk mengidentifikasi pola data yang tidak sesuai pada atribut *Tag* dan *Host*. Pada atribut *Tag*, Regex digunakan untuk mengenali nilai yang memiliki pola *timestamp*, sedangkan pada atribut *Host* digunakan untuk mengidentifikasi nilai berupa nama bulan yang tidak sesuai dengan karakteristik atribut tersebut. Selain melakukan pembersihan, Regex juga digunakan untuk mengekstraksi informasi dari atribut *Message* seperti *MAC Address* yang kemudian dimanfaatkan untuk melengkapi nilai *Host* yang kosong. Hal tersebut menunjukkan bahwa penerapan Regex pada penelitian ini tidak hanya digunakan untuk menghilangkan data yang tidak sesuai, tetapi juga untuk memanfaatkan kembali informasi yang terdapat pada data log.

Pada proses *log parsing*, Regex diterapkan pada atribut *Message* yang memiliki struktur *Common Event Format* (CEF). Pola Regex yang digunakan memisahkan isi pesan berdasarkan delimiter `|` sehingga menghasilkan tujuh komponen informasi, yaitu *CEF Version*, *Vendor*, *Product*, *Version*, *Process*, *Event*, dan *Severity*. Informasi hasil *parsing* tersebut kemudian digunakan untuk memperoleh nama perangkat dari bagian *Event* dan melengkapi atribut *Host*. Selain itu, beberapa informasi CEF digunakan untuk membentuk nilai *Tag* pada data yang memenuhi kondisi tertentu. Dengan demikian, proses *log parsing* membantu mengubah informasi yang sebelumnya tersimpan dalam bentuk teks menjadi struktur data yang lebih mudah dikelola dan digunakan dalam proses pengolahan berikutnya.

Penerapan Regex dalam penelitian ini memiliki kesamaan dengan penelitian terdahulu yang menggunakan Regex untuk mengidentifikasi dan mengekstraksi pola tertentu dari data berbasis teks. Penelitian Yogi et al., (2019) menerapkan *Regular Expressions* untuk menganalisis pola perilaku pengunjung pada log web server, sedangkan penelitian Supartini (2023) menggunakan Regex untuk mengidentifikasi pola yang berkaitan dengan serangan *SQL injection*. Penelitian Meiliyani et al., (2025) juga menggunakan Regex untuk mendukung proses pencarian informasi pada aplikasi *e-commerce*. Perbedaan penelitian ini terletak pada objek dan tujuan penerapannya, yaitu Regex digunakan sebagai bagian dari proses *data cleaning* dan *log parsing* pada data *Syslog*, termasuk melakukan normalisasi atribut, mengekstraksi informasi dari *Message*, serta melakukan *parsing* terhadap log dengan format CEF.

Salah satu kelebihan Regex dalam penelitian ini adalah kemampuannya untuk mengenali pola data secara spesifik berdasarkan struktur yang telah diketahui. Regex dapat diterapkan pada beberapa kebutuhan pengolahan dengan pola yang berbeda, mulai dari identifikasi *timestamp*, pembersihan nilai pada atribut *Host*, ekstraksi *MAC Address*, hingga *parsing* pesan CEF. Pendekatan tersebut memungkinkan proses pembersihan dan ekstraksi dilakukan secara terstruktur berdasarkan aturan yang telah ditentukan. Selain itu, penggunaan Regex memungkinkan proses *data cleaning* dan ekstraksi informasi dilakukan dalam satu rangkaian pengolahan tanpa harus memeriksa setiap data secara manual. Meskipun demikian, penggunaan Regex memiliki keterbatasan. Pola yang digunakan dalam penelitian ini dibuat berdasarkan karakteristik *dataset* dan format log yang ditemukan selama proses penelitian. Apabila struktur atau format log mengalami perubahan, pola Regex yang digunakan perlu disesuaikan agar tetap dapat mengenali data dengan benar. Kondisi tersebut menunjukkan bahwa Regex lebih sesuai digunakan pada data yang memiliki pola relatif jelas dan telah diketahui sebelumnya. Sebagai perbandingan, pendekatan log parsing berbasis frekuensi kata seperti PatCluster mampu membentuk pola log secara otomatis tanpa bergantung pada definisi pola secara manual, sehingga lebih adaptif terhadap perubahan struktur log (Bai et al., 2023). Namun demikian, pendekatan semacam ini memerlukan proses komputasi tambahan dan belum tentu memberikan tingkat interpretabilitas yang sama dengan pola Regex yang eksplisit. Untuk lingkungan dengan berbagai format log yang berubah secara dinamis atau memiliki struktur yang sangat kompleks, diperlukan pendekatan tambahan agar proses parsing dapat beradaptasi terhadap perubahan tersebut.

Hasil penelitian juga memiliki implikasi terhadap proses pengelolaan dan monitoring data *Syslog*. Data yang telah melalui proses *cleaning* dan *parsing* memiliki struktur yang lebih konsisten sehingga dapat memudahkan proses pencarian, penyaringan, dan penelusuran informasi pada sistem berbasis web. Informasi yang terdapat pada atribut *Message* juga dapat dimanfaatkan untuk melengkapi atribut lain sehingga data menjadi lebih informatif. Bagi administrator jaringan, hasil pengolahan tersebut dapat menjadi tahap awal dalam menyiapkan data log yang lebih terstruktur untuk mendukung proses monitoring dan analisis keamanan jaringan. Namun, penelitian ini belum menerapkan metode deteksi anomali atau deteksi serangan secara otomatis, sehingga hasil penelitian tidak dapat digunakan untuk menyatakan kemampuan sistem dalam mendeteksi ancaman secara otomatis. Posisi penelitian ini lebih tepat sebagai tahap *data preprocessing*, *data cleaning*, dan *log parsing* yang mendukung proses monitoring keamanan jaringan.

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, penerapan *Regular Expression* (Regex) pada data *Syslog* dapat digunakan untuk mendukung proses *data cleaning* dan *log parsing* sehingga data menjadi lebih terstruktur dan konsisten. Proses *data cleaning* dilakukan pada atribut *Tag*, *Time*, dan *Host* yang meliputi identifikasi dan pembersihan nilai yang tidak sesuai, normalisasi format waktu, serta pemanfaatan informasi dari atribut *Message* untuk melengkapi data yang kosong. Hasil pengolahan menunjukkan bahwa Regex dapat digunakan untuk mengenali pola tertentu dan mengekstraksi informasi yang diperlukan sesuai dengan karakteristik data *Syslog* yang digunakan dalam penelitian. Pada proses *log parsing*, Regex digunakan untuk mengolah pesan dengan struktur *Common Event Format* (CEF) dan memisahkannya menjadi tujuh komponen yaitu *CEF Version*, *Vendor*, *Product*, *Version*, *Process*,

Event, dan *Severity*. Informasi hasil *parsing* kemudian digunakan untuk melengkapi atribut *Host* dan *Tag*, serta memperoleh informasi nama perangkat dari bagian *Event*. Proses tersebut menghasilkan data log yang lebih terstruktur sehingga lebih mudah dikelola dan digunakan pada tahap pengolahan berikutnya.

Data yang telah melalui *data cleaning* dan *log parsing* selanjutnya disimpan ke dalam MySQL sebagai basis data. Sistem berbasis web yang dikembangkan menyediakan fitur *dashboard*, *upload file*, *search*, dan filter berdasarkan rentang waktu untuk membantu pengguna mengelola dan menelusuri data *Syslog* yang telah diproses. Dengan adanya integrasi antara proses pengolahan data, penyimpanan basis data, dan antarmuka web, hasil penelitian dapat mendukung pengelolaan serta monitoring data *Syslog* secara lebih terstruktur. Meskipun demikian, penerapan Regex pada penelitian ini masih bergantung pada pola yang telah ditentukan berdasarkan karakteristik dataset yang digunakan. Perubahan format atau struktur log dapat memerlukan penyesuaian terhadap pola Regex. Selain itu, penelitian ini belum menerapkan metode deteksi anomali atau deteksi serangan secara otomatis. Oleh karena itu, hasil penelitian ini lebih tepat diposisikan sebagai tahap awal data preprocessing dan analisis data *Syslog* yang mendukung proses monitoring keamanan jaringan, bukan sebagai sistem deteksi ancaman otomatis.

DAFTAR PUSTAKA

- Anton, A. A., Csereoka, P., Capota, E. A., & Cioargă, R. D. (2024). Enhancing Syslog Message Security and Reliability over Unidirectional Fiber Optics. *Sensors*, 24(20), 1–25. <https://doi.org/10.3390/s24206537>
- Aulia, D., Telaumbanua, I. A. A., Putri, W. F., Wijaya, M. A., & Nurbaiti. (2023). Efektivitas Sistem Database Di Dalam Sistem Informasi Manajemen. *Jurnal Ilmiah Sains Teknologi Dan Informasi*, 1(3), 27–34. <https://doi.org/10.59024/jiti.v1i3.283>
- Bai, Y., Chi, Y., & Zhao, D. (2023). PatCluster: A Top-Down Log Parsing Method Based on Frequent Words. *IEEE Access*, 11, 8275–8282. <https://doi.org/10.1109/ACCESS.2023.3239012>
- Bintang, J. M., Ashshidiq, M. F., & Dzakwan, H. F. (2023). Penerapan Algoritma String Matching dan Regular Expression pada Aplikasi Kamus Besar Bahasa Indonesia (KBBI). *BIOS Jurnal Teknologi Informasi dan Rekayasa Komputer*, 4(1), 34–41. <https://doi.org/10.37148/bios.v4i1.57>
- Cahyo, A. N., Hermawan, R., Wijaya, M. A. D., & Sari, A. P. (2025). Deteksi Teks Promosi Judi Online Pada Kolom Komentar YouTube dengan Metode Regex dan Fuzzy Matching. *Jurnal RESTIKOM: Riset Teknik Informatika Dan Komputer*, 7(2), 176–187. <https://doi.org/10.52005/restikom.v7i2.441>
- Chuah, E., Kalutarage, H., Tasdemir, K., Abrham, A., & Maple, C. (2025). A Systematic Literature Review of Log-Correlation Tools for Cyberattack Detection and Prediction in Large Networks. *Journal of Information Security and Applications*, 92, 1–39. <https://doi.org/10.1016/j.jisa.2025.104096>
- Hamid, M. F. (2022). Penanganan Situs Internet Bermuatan Negatif. *DHARMASISYA Jurnal Program Magister Fakultas Hukum Universitas Indonesia*, 2(2), 865–880.
- He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An Online Log Parsing Approach with Fixed Depth Tree. *IEEE International Conference on Web Services (ICWS)*, 33–40. <https://doi.org/10.1109/ICWS.2017.13>
- Huang, S., Wu, X., & Li, H. (2025). LogLSHD: Fast Log Parsing with Locality-Sensitive Hashing and Dynamic Time Warping. *Companion Proceedings of the 33rd ACM Symposium on the Foundations of Software Engineering (FSE '25)*, 1–10. <https://doi.org/10.48550/arXiv.2504.02172>
- Junaedi, J., Maranto, A. R. K., Putri, M. P., & Suwitno. (2023). User Interface Experience Analysis of PMB Online Buddhi Dharma Using System Usability Scale. *Bit-Tech (Binary Digital-Technology)*, 6(2), 216–222. <https://doi.org/10.32877/bt.v6i2.1051>
- Liu, C., Pavlenko, A., Interlandi, M., & Haynes, B. (2025). Data Formats in Analytical DBMSs :

- Performance Trade-Offs And Future Directions. *The VLDB Journal*, 34(3), 1–23. <https://doi.org/10.1007/s00778-025-00911-1>
- Maspupah, A. (2022). Literature Review: Advantages And Disadvantages Of Black Box And White Box Testing Methods. *Techno Nusa Mandiri: Journal of Computing and Information Technology*, 21(2), 151–162. <https://doi.org/10.33480/techno.v21i2.5776>
- Meiliyani, K., Alfah, Awalia, N., & Setiadi, T. (2025). Penggunaan Ekspresi Regular dalam Meningkatkan Efisiensi Pencarian Produk pada Aplikasi E-Commerce. *Neptunus: Jurnal Ilmu Komputer Dan Teknologi Informasi*, 3(1), 169–177. <https://doi.org/10.61132/neptunus.v3i1.698>
- Ramadhan, F. H., & Putro, H. P. (2020). Implementasi DataTables pada Proyek The Point Of Sale. *AUTOMATA*, 1(2), 20–44.
- Saifudin, A., Mulyati, S., Sidi, R. G., Tanjung, R. F., Hermawan, I., & Ruziki, N. N. (2022). Pengujian Black Box pada Aplikasi Penjualan Toko Bunga Pelangi Berbasis Web Menggunakan Metode Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 7(1), 138–144. <https://doi.org/10.32493/informatika.v7i1.17523> 3HQJXMLDQ
- Salih, Y., & Saefullah, R. (2024). Black Box Testing on Website-Based Guestbook Registration Applications. *International Journal of Mathematics, Statistics, and Computing*, 2(2), 44–49. <https://doi.org/10.46336/ijmsc.v2i2.96>
- Salunke, S. V., & Ouda, A. (2024). A Performance Benchmark for the PostgreSQL and MySQL Databases. *Future Internet*. 16(10). <https://doi.org/10.3390/fi16100382>
- Sarker, I. H. (2021). Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective. *SN Computer Science*, 2(5), 1–22. <https://doi.org/10.1007/s42979-021-00765-8>
- Sasidharan, R. (2022). Implementation of High Available and Scalable Syslog Server with NoSQL Cassandra Database and Message Queue. *American Journal of Computer Architecture*, 9(1), 1–7. <https://doi.org/10.5923/j.ajca.20220901.01>
- Sedki, I., Hamou-lhadj, A., & Ait-mohamed, O. (2021). AWSOM-LP: An Effective Log Parsing Technique Using Pattern Recognition and Frequency Analysis. *ArXiv*, 1–13. <https://doi.org/10.48550/arXiv.2110.15473>
- Sheeraz, M., Durad, M. H., Paracha, M. A., Mohsin, S. M., Kazmi, S. N., & Maple, C. (2024). Revolutionizing SIEM Security: An Innovative Correlation Engine Design for Multi-Layered Attack Detection. *Sensors*, 24(15), 1–22. <https://doi.org/10.3390/s24154901>
- Supartini, R. (2023). Deteksi Serangan SQL Injection pada Website dengan Menggunakan Metode Regular Expression. *Progressive Information, Security, Computer, and Embedded System*, 1(2), 107–114. <https://doi.org/10.61255/pisces.v1i2.101>
- Teneke, N. G., Etemi, J. G., Kamak, Y. S., & Ariwa, R. N. (2024). Smart Library Automation: Integrating Classification Tree-Based cataloging with XAMPP Server Environment. *ABUAD Journal of Engineering Research and Development (AJERD)*, 7(1), 328–339. <https://doi.org/10.53982/ajer.2024.0701.33-j>
- Yogi, Ruslianto, I., & Bahri, S. (2019). Analisa Log Web Server Untuk Mengetahui Pola Perilaku Pengunjung Website Menggunakan Teknik Regular Expressions. *Coding Jurnal Komputer Dan Aplikasi*, 7(01), 120–130. <https://doi.org/10.26418/coding.v7i01.32692>
- Zhang, T., Qiu, H., Castellano, G., Rifai, M., Chen, C. S., & Pianese, F. (2023). System Log Parsing: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(8), 8596–8614. <https://doi.org/10.1109/TKDE.2022.3222417>