Pemanfaatan Regular Expression untuk Deteksi Pola Serangan pada Log Jaringan

Muhammad Iqbal Haidar - 13523111
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: haidariqbalm@gmail.com , 13523111@std.stei.itb.ac.id

***Abstrak*—Makalah ini membahas pemanfaatan regular expression (regex) untuk mendeteksi pola serangan pada log jaringan, dengan fokus pada serangan SQL Injection, Cross-site Scripting (XSS), dan Command Injection. Program dikembangkan menggunakan Python dan dirancang untuk membaca log secara baris per baris, kemudian mencocokkan setiap baris dengan pola regex yang telah ditentukan. Dataset yang digunakan berupa data log sintetis yang meniru struktur log server web, dengan penyisipan pola-pola serangan secara acak. Hasil implementasi menunjukkan bahwa pendekatan regex mampu secara efektif mengidentifikasi entri log yang mengandung indikasi serangan, serta menyajikan keluaran yang jelas dan informatif. Statistik jumlah deteksi per jenis serangan juga memberikan gambaran mengenai sebaran ancaman dalam dataset. Dengan fleksibilitas tinggi dan kemudahan pengembangan, pendekatan ini dapat dijadikan solusi awal dalam proses analisis keamanan berbasis log jaringan.**

Kata Kunci—Pattern Matching, Regular Expression, Keamanan Jaringan

I. PENDAHULUAN

Di era digital saat ini, pertukaran informasi menjadi hal penting yang dapat mendukung kegiatan manusia sehari-hari. Setiap harinya lebih dari triliunan byte data ditransmisikan dari suatu tempat ke tempat lain melalui jaringan internet. Semua informasi tersebut bisa didapatkan dalam genggaman tangan berkat perkembangan teknologi informasi. Oleh sebab itu, teknologi informasi memiliki peranan besar dalam mendorong umat manusia berakselerasi menuju peradaban maju. Di dunia dengan perkembangan teknologi yang pesat seperti saat ini, inovasi dalam bidang teknologi informasi selalu dinantikan oleh khalayak ramai sebab implementasinya dapat dirasakan secara universal, tidak terbatas pada golongan tertentu saja. Hal tersebut mendorong para ilmuwan serta perusahaan teknologi informasi berlomba-lomba menciptakan inovasi baru yang dapat merevolusi bidang ini untuk perkembangan umat manusia.

Jaringan komputer adalah cabang ilmu yang mempelajari hubungan komunikasi antara perangkat komputer dengan perangkat komputer lainnya. Ilmu ini mendasari terbentuknya jaringan internet, yakni jaringan yang menghubungkan perangkat komputer di seluruh belahan dunia dan dapat diakses secara publik oleh semua orang. Lebih dari itu ada juga jaringan intranet yang banyak digunakan oleh perusahaan ataupun pemerintahan sebagai sarana berbagi data yang sifatnya tidak boleh diketahui publik.

Seiring dengan perkembangan pesat dunia teknologi informasi tentu saja tidak bisa lepas dari oknum-oknum jahat yang mencoba memanfaatkan kelemahan-kelemahan yang ada guna di eksploitasi untuk kepentingan dirinya sendiri. Motif dibaliknya beragam, ada yang didorong karena faktor ekonomi, keamanan, atau bahkan sekadar untuk bersenang-senang. Target operasi dari oknum-oknum tersebut pun bervariasi yakni pemerintahan, korporasi, atau bahkan kaum masyarakat rentan seperti lansia dan anak kecil. Oknum-oknum ini akan melakukan berbagai metode untuk mendapatkan hal-hal yang diinginkan dari targetnya, mulai dari melakukan penetrasi jaringan, menyebarkan virus komputer, bahkan sampai berpura-pura untuk menjadi orang lain agar bisa mendapatkan akses yang lebih tinggi terhadap sistem yang akan diserang. Sehingga diperlukan untuk melakukan tindakan preventif guna mencegah ataupun memperkecil peluang seseorang untuk menyerang sistem yang ada.

Salah satu cara untuk melakukan tindakan preventif tersebut adalah dengan melakukan analisis terhadap log jaringan dari sistem tersebut. Analisis jaringan dilakukan dengan mencari kata kunci atau pola serangan yang umum terjadi ketika seseorang melakukan percobaan memasuki dan mengubah sistem tanpa akses yang sesuai. Dengan begitu admin jaringan akan lebih mudah untuk melakukan pencarian serta penelusuran terhadap seseorang yang dicurigai sedang melakukan serangan terhadap sistem. Lebih dari itu, dengan proses otomasi yang baik maka dapat dijadikan sebagai sistem pengawasan terhadap jaringan guna memastikan aspek integritasnya secara keseluruhan. Salah satu alternatif untuk mengimplementasikan sistem tersebut adalah dengan memanfaatkan regular expression.

Makalah ini akan membahas bagaimana mengimplementasikan regular expression untuk melakukan analisis log jaringan dari sebuah sistem. Fokus utama pembahasan adalah bagaimana pola-pola serangan atau aktivitas mencurigakan seperti percobaan login yang berulang, akses ke endpoint sensitif, hingga penyisipan perintah-perintah berbahaya seperti SQL Injection dapat dikenali secara otomatis dengan pencocokan pola menggunakan regex. Melalui pendekatan ini, diharapkan pembaca dapat memahami peran strategi algoritma pencocokan string dan regular expression sebagai solusi praktis dalam keamanan jaringan, khususnya dalam sistem deteksi dini dan pengawasan log otomatis..

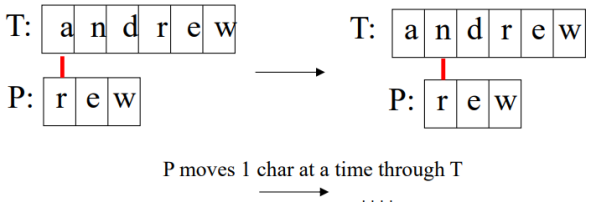
II. LANDASAN TEORI

A. *Pattern Matching*

Pattern matching adalah sebuah algoritma dimana diberikan sebuah teks panjang (T) berupa string dengan panjang totalnya mencapai n karakter lalu akan dicari sebuah pattern (P) sepanjang m karakter dengan mengasumsikan bahwa panjang T >> P dalam teks tersebut. Setelahnya akan dikembalikan kemunculan pertama pattern P dalam teks T yang umumnya berupa indeks karakter pertama dari pattern P pada kemunculan pertama di teks T. Apabila pattern tidak ditemukan dalam teks T maka akan mengembalikan indeks tidak valid yakni -1. Dalam implementasinya terdapat berbagai macam algoritma pattern matching, beberapa yang umum untuk digunakan antara lain adalah Brute Force, Knuth-Morris-Pratt (KMP), Boyer-Moore (BM), dan Aho-Corasick (AC). Selain itu, ekspresi pattern yang digunakan juga dapat menggunakan Regular Expression sehingga lebih fleksibel. Algoritma pattern matching banyak digunakan dalam kehidupan sehari-hari seperti web engine, analisis citra digital, dan bioinformatika.

B. *Brute Force*

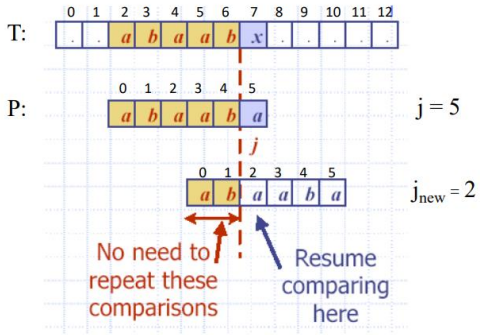
Brute force merupakan algoritma pattern matching yang paling sederhana dan mudah untuk dimengerti serta diimplementasikan. Namun algoritma ini memiliki kekurangan yakni dari sisi kompleksitasnya yang tidak sebaik algoritma pattern matching lainnya. Pada dasarnya algoritma brute force memiliki cara kerja yang cukup simpel yakni melakukan pengecekan di setiap karakter pada pattern untuk melihat apakah pattern P dimulai dari indeks awal pada teks tersebut. Kompleksitas waktu yang dimiliki oleh algoritma ini untuk kasus terburuk adalah O(mn) sedangkan untuk kasus terbaik adalah O(n) dan kasus rata-rata adalah O(m + n). Algoritma ini bekerja lebih baik ketika teks yang akan dilakukan pattern matching memiliki variasi karakter yang besar, dan sebaliknya bekerja lebih buruk ketika variasi karakter teks kecil misalnya seperti pada teks biner.



Gambar 2.1 Algoritma Pattern Matching Brute Force
Sumber:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf

C. *Knuth-Morris-Pratt (KMP)*

Knuth-Morris-Pratt adalah algoritma pattern matching yang dapat dibilang merupakan improvement dari algoritma Brute Force. Algoritma ini memiliki ide dasar mirip seperti Brute Force yakni melakukan pengecekan dari kiri ke kanan satu per satu untuk setiap karakternya. Namun diberikan tambahan yakni ketika terjadi ketidakcocokan karakter pada indeks ke-x maka berapa banyak pergeseran pattern terhadap teks yang bisa dilakukan agar tidak terjadi proses pengecekan berulang yang sia-sia. Pertanyaan tersebut dijawab oleh algoritma ini dengan menyatakan bahwa pergeseran yang dilakukan adalah sebesar selisih jumlah karakter pattern P dengan jumlah karakter prefiks dari P[0..j-1] yang juga merupakan suffiks dari P[1..j-1]. Kompleksitas waktu yang dimiliki oleh algoritma ini adalah O(m + n), jauh lebih cepat ketimbang algoritma brute force. Namun algoritma ini memiliki kekurangan yakni ketika variansi dari karakter pada teks besar, maka algoritma ini menjadi kurang efektif karena semakin besar potensi mismatch saat proses pencarian.



Gambar 2.2 Algoritma Pattern Matching KMP
Sumber:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf

D. *Boyer-Moore (BM)*

Boyer-Moore adalah algoritma pattern matching dengan memanfaatkan dua teknik utama yakni

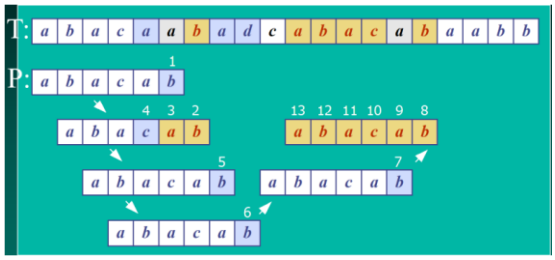
1. Teknik Looking-Glass

   Mencari pattern P pada teks T dengan melakukan pengecekan setiap karakter pada pattern P secara mundur atau dari arah kanan ke kiri.

2. Teknik Character-Jump

   Ketika saat pencarian terjadi missmatch maka akan terbagi menjadi tiga kemungkinan yang akan dicoba secara berurutan sesuai dengan yang termasuk pada kondisionalnya.

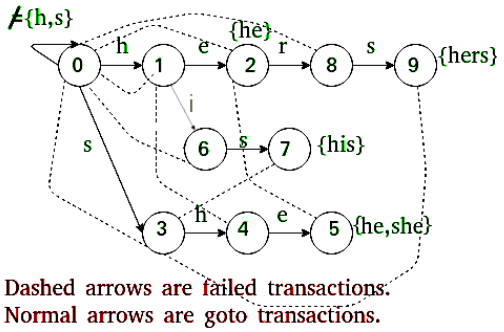
Kompleksitas algoritma yang dimiliki oleh algoritma ini pada kasus terburuknya adalah O(mn + A), ini lebih baik ketimbang algoritma pattern matching lain yakni brute-force. Algoritma ini berjalan dengan optimal ketika variansi karakter pada teksnya besar sehingga cocok digunakan untuk pencarian teks berbahasa inggris, namun kurang cocok untuk digunakan pada algoritma pencarian teks biner.



Gambar 2.3 Algoritma Pattern Matching BM
Sumber:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf

E. *Aho-Corasick*

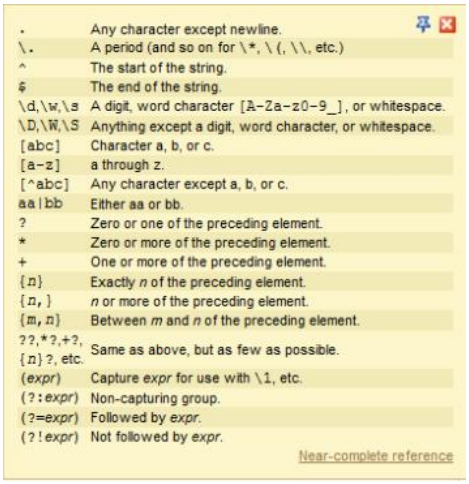
Algoritma Aho-Corasick adalah algoritma pencocokan pola (pattern matching) yang efisien untuk menemukan banyak pola sekaligus dalam sebuah teks. Algoritma ini bekerja dengan membangun struktur data berupa automaton atau trie yang mewakili seluruh kumpulan pola yang ingin dicocokkan. Setelah struktur ini dibentuk, algoritma akan memproses teks masukan satu per satu karakter sambil mengikuti transisi dalam trie dan menggunakan fungsi "failure link" untuk menangani ketidaksesuaian secara efisien, mirip dengan algoritma KMP. Dengan demikian, Aho-Corasick mampu melakukan pencarian semua pola dengan kompleksitas O(mk) dimana m merupakan string penyusun dan k adalah besar alfabetnya. Karena kemampuannya mencocokkan banyak pola sekaligus dengan efisiensi tinggi, algoritma ini banyak gunakan dalam aplikasi seperti penyaringan konten, pendeteksian virus, dan analisis log.



Gambar 2.4 Algoritma Pattern Matching Aho-Corasick
Sumber: https://www.geeksforgeeks.org/dsa/aho-corasick-algorithm-pattern-searching/

F. *Regular Expression (regex)*

Regular expression (regex) adalah suatu pola atau rangkaian karakter yang digunakan untuk mencocokkan, mencari, dan memanipulasi teks berdasarkan aturan yang telah ditentukan. Regex memungkinkan pengguna untuk mengekspresikan pencarian string dengan cara yang sangat fleksibel dan efisien, sehingga sering digunakan dalam pemrograman, analisis data, serta pengolahan teks. Dengan regex, kita bisa mendeteksi pola tertentu seperti email, nomor telepon, tanggal, atau bahkan kata-kata tertentu dalam dokumen besar dengan cepat. Pola ini disusun menggunakan karakter biasa dan simbol khusus seperti . untuk mencocokkan satu karakter apa saja, * untuk menyatakan pengulangan, [] untuk menyatakan himpunan karakter, dan ^ atau $ untuk menyatakan awal atau akhir baris. Karena kemampuannya yang kuat dan luas, regex menjadi alat penting dalam berbagai bidang seperti keamanan siber, pemrosesan bahasa alami, hingga validasi input dalam aplikasi web dan perangkat lunak.



Gambar 2.5 Notasi Umum Regular Expression
Sumber:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf

G. *Keamanan Jaringan*

Keamanan jaringan adalah praktik melindungi jaringan komputer dari akses tidak sah, penyalahgunaan, atau serangan. Ini mencakup kombinasi dari perangkat lunak, perangkat keras, kebijakan, prosedur, dan pengaturan untuk menjaga kerahasiaan, integritas, dan ketersediaan data dalam jaringan. Tingkat keamanan jaringan dibagi menjadi tiga yakni:

1. Keamanan Fisik – Mencegah akses fisik oleh pihak tidak berwenang (misalnya melalui biometrik)
2. Keamanan Teknis – melindungi data dalam penyimpanan dan selama transmisi
3. Keamanan Administratif – mengatur perilaku pengguna dan proses otorisasi dalam jaringan

Beberapa teknik penyerangan yang umum dilakukan pada sebuah jaringan antara lain adalah:

1. SQL Injection

   SQL Injection adalah salah satu jenis serangan keamanan yang memanfaatkan celah pada aplikasi yang berinteraksi dengan basis data, di mana penyerang menyisipkan perintah SQL berbahaya ke dalam input pengguna. Tujuan dari serangan ini adalah untuk memanipulasi query database agar dapat mengakses, mengubah, atau menghapus data yang seharusnya tidak dapat diakses tanpa izin. SQL Injection biasanya terjadi ketika input tidak divalidasi dengan baik, seperti pada kolom login atau parameter URL, sehingga perintah SQL seperti ' OR 1=1 -- dapat dijalankan langsung oleh sistem.

2. Cross Site Scripting (XSS)

   Cross-site Scripting (XSS) adalah jenis serangan keamanan pada aplikasi web di mana penyerang menyisipkan skrip berbahaya ke dalam halaman web yang kemudian dijalankan di sisi pengguna (browser). Serangan ini biasanya terjadi ketika aplikasi web menampilkan input pengguna tanpa validasi atau

penyaringan yang memadai, memungkinkan penyerang mengirimkan kode JavaScript yang dapat mencuri data sensitif, seperti cookie, token sesi, atau memanipulasi tampilan halaman. XSS sering terjadi melalui form input, URL, atau kolom komentar yang tidak diamankan dengan benar.

3. Brute Force Attack

   Brute force attack adalah jenis serangan di mana penyerang mencoba menebak kredensial, seperti username dan password, dengan mencoba semua kemungkinan kombinasi secara sistematis hingga menemukan yang benar. Serangan ini biasanya dilakukan secara otomatis menggunakan skrip atau program, dan dapat memakan waktu lama tergantung pada panjang dan kompleksitas data yang ditebak. Brute force sering digunakan untuk membobol akun login yang lemah keamanannya, terutama jika tidak ada batasan percobaan atau mekanisme penguncian setelah sejumlah kegagalan.

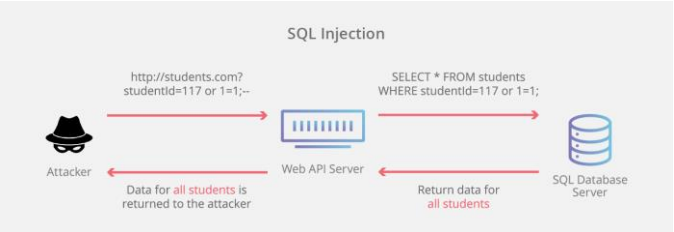
4. Distributed Denial of Service (DDoS)

   Distributed Denial of Service (DDoS) adalah jenis serangan siber di mana penyerang membanjiri sebuah server, layanan, atau jaringan dengan lalu lintas yang sangat besar secara bersamaan dari banyak sumber (komputer atau bot), dengan tujuan membuat layanan menjadi lambat atau tidak dapat diakses oleh pengguna yang sah. Serangan ini memanfaatkan banyak perangkat yang telah dikompromikan (botnet) untuk mengirim permintaan secara terus-menerus, sehingga server korban kehabisan sumber daya dan gagal melayani permintaan normal. DDoS merupakan salah satu serangan yang sulit ditangani karena lalu lintas serangan berasal dari berbagai lokasi yang tersebar.

5. Phising

   Phishing adalah jenis serangan siber yang bertujuan untuk menipu korban agar memberikan informasi sensitif seperti username, password, atau data kartu kredit dengan menyamar sebagai entitas tepercaya. Serangan ini biasanya dilakukan melalui email, pesan instan, atau situs web palsu yang dirancang menyerupai layanan resmi. Korban yang tertipu akan memasukkan data pribadinya ke dalam formulir palsu, yang kemudian dikumpulkan oleh penyerang untuk digunakan dalam tindakan penipuan atau pencurian identitas.

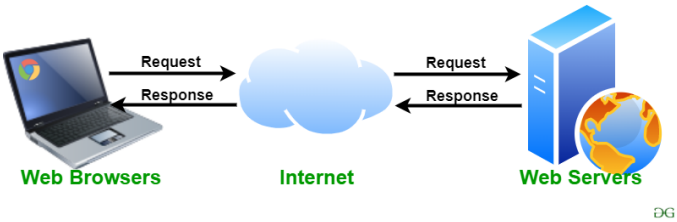
Keamanan jaringan sangat penting untuk melindungi data, sistem, dan komunikasi dalam suatu organisasi. Dengan penerapan berlapis seperti firewall, enkripsi, dan IPS, serta edukasi pengguna, risiko kebocoran dan serangan dapat diminimalisasi. Strategi keamanan yang baik memastikan jaringan tetap aman, andal, dan terlindungi dari berbagai ancaman siber.



Gambar 2.6 Serangan Jaringan: SQL Injection
Sumber: https://www.cloudflare.com/learning/security/threats/sql-injection/

H. Web Server

Web server adalah perangkat lunak atau perangkat keras yang berfungsi untuk menerima, memproses, dan merespons permintaan dari klien melalui protokol HTTP atau HTTPS, biasanya dalam bentuk halaman web. Ketika pengguna mengakses sebuah situs melalui browser, permintaan dikirim ke web server, yang kemudian mencari konten yang diminta seperti file HTML, gambar, atau data dari database dan mengirimkannya kembali ke browser untuk ditampilkan. Web server juga bisa menjalankan skrip atau aplikasi web dinamis, seperti PHP atau JavaScript, untuk menghasilkan konten secara real-time. Contoh web server yang populer termasuk Apache, Nginx, dan Microsoft IIS. Web server berperan penting dalam infrastruktur internet karena menjadi penghubung utama antara pengguna dan aplikasi berbasis web.



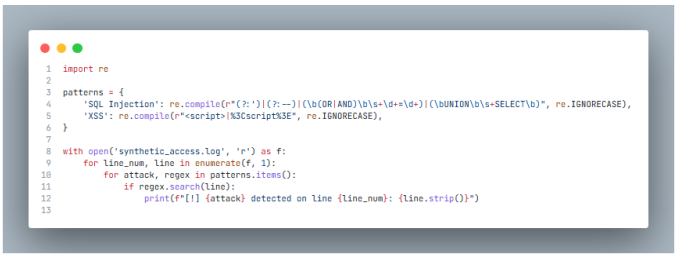
Gambar 2.7 Diagram Web Server
Sumber: https://www.geeksforgeeks.org/node-js/web-server-and-its-type/

III. IMPLEMENTASI

Program ini dikembangkan menggunakan bahasa pemrograman Python versi 3.x karena kemudahan sintaksisnya serta dukungan pustaka bawaan yang kuat, terutama modul re untuk pengolahan regular expression. Program ini dirancang untuk melakukan analisis terhadap file log jaringan dalam format akses HTTP, dengan fokus utama pada pendeteksian tiga jenis serangan yang umum terjadi, yaitu SQL Injection, Cross-site Scripting (XSS), dan Command Injection. Setiap jenis serangan memiliki pola unik yang dapat dicirikan oleh serangkaian ekspresi karakteristik, seperti penggunaan tanda kutip tunggal ('), perintah UNION SELECT, skrip <script>, atau karakter pemisah perintah seperti ; dan &&. Pola-pola ini kemudian diformulasikan menjadi ekspresi reguler (regex) yang dapat secara otomatis mencocokkan baris-baris log yang mencurigakan berdasarkan konten string-nya.

Struktur program dibangun secara sederhana namun efektif, dimulai dari pembacaan file log baris per baris, kemudian dilakukan proses pencocokan pola terhadap setiap baris. Jika

ditemukan kecocokan terhadap salah satu pola serangan, program akan mencetak informasi deteksi berupa jenis serangan dan isi baris log yang mencurigakan. Implementasi ini bersifat modular dan dapat dikembangkan lebih lanjut untuk mendeteksi jenis serangan lain hanya dengan menambahkan pola regex yang relevan.



```python
import re

patterns = {
    'SQL Injection': re.compile(r"('|(?:--)|(\b(OR|AND)\b\s+\d+=\d+)|(\bUNION\b\s+SELECT\b)", re.IGNORECASE),
    'XSS': re.compile(r"<script>|%3Cscript%3E", re.IGNORECASE),
}

with open('synthetic_access.log', 'r') as f:
    for line_num, line in enumerate(f, 1):
        for attack, regex in patterns.items():
            if regex.search(line):
                print(f"[!] {attack} detected on line {line_num}: {line.strip()}")
```

Gambar 2.8 Cuplikan Program
Sumber: Arsip Penulis

Dataset yang digunakan dalam penelitian ini merupakan dataset sintetis yang dibuat untuk menyerupai struktur log akses server web seperti Apache. Dataset ini terdiri dari 80 baris data log, yang sebagian besar merupakan permintaan HTTP normal, dan 10 di antaranya mengandung pola-pola yang mencerminkan serangan siber. Format log mencakup informasi alamat IP, timestamp, metode permintaan HTTP, endpoint URL, status HTTP, serta ukuran respons. Pola serangan disisipkan secara acak di antara log normal untuk mensimulasikan kondisi log jaringan nyata yang mengandung aktivitas mencurigakan.

```
192.168.1.1 - - [12/Jun/2025:10:21:26 +0700] "POST /contact HTTP/1.1" 301 1933 "curl/7.68.0"
203.0.113.10 - - [12/Jun/2025:10:27:51 +0700] "GET /contact HTTP/1.1" 200 2110 "curl/7.68.0"
172.16.0.3 - - [12/Jun/2025:10:58:18 +0700] "GET /admin HTTP/1.1" 404 1072 "Mozilla/5.0"
192.168.1.1 - - [12/Jun/2025:10:49:25 +0700] "GET /login HTTP/1.1" 500 4185 "Mozilla/5.0"
192.168.1.1 - - [12/Jun/2025:10:31:01 +0700] "POST /admin HTTP/1.1" 200 606 "PostmanRuntime/7.29.2"
10.0.0.2 - - [12/Jun/2025:10:20:02 +0700] "POST /search HTTP/1.1" 301 3452 "curl/7.68.0"
192.168.1.1 - - [12/Jun/2025:10:26:17 +0700] "GET /contact HTTP/1.1" 200 3107 "PostmanRuntime/7.29.2"
203.0.113.10 - - [12/Jun/2025:10:41:33 +0700] "GET /admin HTTP/1.1" 500 602 "curl/7.68.0"
172.16.0.3 - - [12/Jun/2025:10:31:11 +0700] "GET /contact HTTP/1.1" 301 4353 "PostmanRuntime/7.29.2"
10.0.0.2 - - [12/Jun/2025:10:04:21 +0700] "GET /search HTTP/1.1" 500 2422 "curl/7.68.0"
```

Gambar 2.9 Cuplikan Dataset
Sumber: Arsip Penulis

Setelah menjalankan program deteksi terhadap dataset log yang telah disiapkan, diperoleh hasil berupa identifikasi baris-baris log yang mengandung pola serangan. Program berhasil mendeteksi sejumlah baris yang mengandung indikasi SQL Injection, XSS, dan Command Injection berdasarkan kecocokan dengan pola-pola regex yang telah dirancang sebelumnya. Deteksi ini menunjukkan bahwa penggunaan regular expression cukup efektif dalam menangkap indikasi serangan berbasis string di dalam log jaringan.

```
[!] XSS detected on line 24: 10.0.0.2 - - [12/Jun/2025:10:52:16 +0700] "GET /search?q=<script>alert(1)</script> HTTP/1.1" 200 1234
[!] XSS detected on line 25: 10.0.0.2 - - [12/Jun/2025:10:24:57 +0700] "GET /search?q=<script>alert(1)</script> HTTP/1.1" 200 1234
[!] SQL Injection detected on line 28: 192.168.1.1 - - [12/Jun/2025:10:26:31 +0700] "GET /login?user=admin'-- HTTP/1.1" 200 1234
[!] SQL Injection detected on line 31: 172.16.0.3 - - [12/Jun/2025:10:40:34 +0700] "POST /search HTTP/1.1" --data="q=1; DROP TABLE users;" 200 1234
[!] SQL Injection detected on line 39: 192.168.1.1 - - [12/Jun/2025:10:50:04 +0700] "GET /products?id=1 UNION SELECT password FROM users HTTP/1.1" 200 1234
[!] SQL Injection detected on line 48: 10.0.0.2 - - [12/Jun/2025:10:02:08 +0700] "POST /search HTTP/1.1" --data="q=1; DROP TABLE users;" 200 1234
[!] SQL Injection detected on line 56: 192.168.1.1 - - [12/Jun/2025:10:17:22 +0700] "GET /products?id=1 UNION SELECT password FROM users HTTP/1.1" 200 1234
[!] SQL Injection detected on line 70: 192.168.1.1 - - [12/Jun/2025:10:38:44 +0700] "GET /login?user=admin'-- HTTP/1.1" 200 1234
```

Gambar 2.10 Cuplikan Hasil
Sumber: Arsip Penulis

Secara statistik, jumlah deteksi untuk masing-masing jenis serangan menunjukkan bahwa SQL Injection merupakan jenis serangan yang paling sering muncul dalam dataset ini, diikuti oleh XSS dan Command Injection. Hal ini sejalan dengan fakta bahwa SQL Injection merupakan salah satu bentuk serangan yang paling umum dan paling mudah dideteksi melalui pola input yang tidak lazim.

Keluaran dari program berupa informasi jenis serangan dan isi baris log yang terdampak, dicetak ke layar konsol. Format keluaran memudahkan analisis lebih lanjut karena menampilkan lokasi baris serta isi asli dari log yang terdeteksi. Selain itu, struktur program memungkinkan untuk dikembangkan agar menyimpan hasil ke dalam file atau menambahkan laporan waktu jika dibutuhkan untuk sistem deteksi waktu nyata.

IV. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan bahwa penggunaan regular expression (regex) merupakan pendekatan yang efektif untuk mendeteksi pola serangan pada log jaringan, khususnya serangan-serangan yang memiliki ciri khas pada pola string seperti SQL Injection, Cross-site Scripting (XSS), dan Command Injection. Dengan merancang pola-pola regex yang spesifik terhadap karakteristik masing-masing jenis serangan, program mampu secara otomatis mengidentifikasi entri log yang mencurigakan tanpa perlu proses parsing yang kompleks. Pendekatan ini juga bersifat ringan dan efisien, karena hanya bergantung pada pemindaian teks berbasis pola tanpa memerlukan integrasi dengan sistem basis data atau analisis semantik yang mendalam.

Selain itu, implementasi yang dibangun dalam penelitian ini menunjukkan bahwa sistem deteksi berbasis regex sangat fleksibel dan mudah dikembangkan. Dengan hanya menambahkan atau memodifikasi ekspresi reguler yang ada, sistem dapat diperluas untuk mengenali jenis serangan lain atau menyesuaikan dengan format log yang berbeda. Dataset sintetis yang digunakan berhasil meniru kondisi nyata dan memberikan skenario pengujian yang cukup representatif. Secara keseluruhan, pendekatan ini dapat menjadi solusi awal yang sederhana namun bermanfaat dalam proses analisis log untuk keperluan keamanan jaringan, baik dalam skala kecil maupun sebagai bagian dari sistem yang lebih besar.

UCAPAN TERIMA KASIH

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas berkat, rahmat, dan hidayat-Nya sehingga penulisan makalah ini bisa terselesaikan. Penulis juga ingin menyampaikan ucapan terima kasih kepada orang tua yang senantiasa memberikan semangat dan dukungan selama proses pengerjaan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T., selaku dosen pengampu mata kuliah Strategi Algoritma (IF2211) kelas K-2, yang telah menjadi pembimbing, memberikan ilmu dan pengetahuan terkait materi sebagai modal dalam mengembangkan makalah ini. Ucapan terima kasih juga disampaikan kepada semua rekan dan pihak lain yang turut mendukung proses penulisan makalah ini, baik secara moral maupun material, walaupun tidak dapat disebutkan satu per satu.

REFERENSI

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf Diakses pada tanggal 24 Juni 2025.

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/24-String-Matching-dengan-Regex-(2025).pdf Diakses pada tanggal 24 Juni 2025.

[3] https://cp-algorithms.com/string/aho_corasick.html Diakses pada tanggal 24 Juni 2025.
[4] https://www.geeksforgeeks.org/dsa/aho-corasick-algorithm-pattern-searching/ Diakses pada tanggal 24 Juni 2025.
[5] https://www.cisco.com/site/us/en/learn/topics/security/what-is-network-security.html Diakses pada tanggal 24 Juni 2025.
[6] https://www.geeksforgeeks.org/network-security/ Diakses pada tanggal 24 Juni 2025.
[7] https://www.cloudflare.com/learning/security/threats/sql-injection/ Diakses pada tanggal 24 Juni 2025.
[8] https://www.geeksforgeeks.org/node-js/web-server-and-its-type/ Diakses pada tanggal 24 Juni 2025.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bekasi, 24 Juni 2025

Muhammad Iqbal Haidar - 13523111