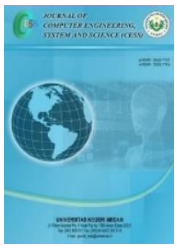
CESS
(Journal of Computing Engineering, System and Science) 8(2) July 2023 607-618

e-ISSN: 2502-714x
p-ISSN: 2502-7131

Contents list available at www.jurnal.unimed.ac.id

CESS
(Journal of Computing Engineering, System and Science)

journal homepage: https://jurnal.unimed.ac.id/2012/index.php/cess



Sistem *Log Analysis* Berbasis Web Untuk Deteksi Serangan Website Menggunakan Algoritma Boyer-Moore Dan Teknik *Regular Expression*

Web-based Log Analysis System for Website Attack Detection using Boyer-Moore Algorithm and Regular Expression Technique

Izzeldin Addarda[1*], Siti Maesaroh[2]

[1,2] Program Studi Teknik Informatika, Universitas Mercu Buana
Jl. Raya Meruya Selatan, Kembangan, Jakarta, DKI Jakarta 11650
email: [1]41519010045@student.mercubuana.ac.id, [2] siti.maesaroh@mercubuana.ac.id

A B S T R A K

Di era digitalisasi saat ini, ancaman serangan siber terus meningkat secara global. serangan siber merujuk pada tindakan kejahatan yang dapat menyebabkan gangguan, pemalsuan dan pencurian informasi berharga dari aplikasi atau situs web. Bersumber pada halaman suara.com terdapat lebih dari 700 juta serangan siber yang berlangsung di Indonesia pada tahun 2022 yang diikuti dengan kebocoran informasi kesehatan e-HAC. Hal tersebut berkaitan dengan peranan dari web server guna melayani permintaan HTTP (Hypertext Transfer Protocol). Selain itu, web server bertugas untuk menerjemahkan kode-kode dinamis menjadi kode-kode statis dalam suatu laman website. Berdasarkan hal tersebut dibuatlah sistem log analysis menggunakan algoritma boyer moore dan teknik regular expressions sebagai metode pencarian adanya indikasi serangan website dengan hasil bahwa algoritma boyer moore dan teknik regular expressions dapat menemukan indikasi jenis serangan terhadap website dengan baik dan relatif cepat karena hanya membutuhkan waktu 32.9 detik untuk menganalisis 1553 baris data yang berasal dari log file. Dengan demikian administrator web server dapat dengan mudah mencari atau melihat jenis upaya serangan website terhadap web server oleh pelaku kejahatan siber.

Kata Kunci: *Algoritma Boyer Moore, Flask, HTTP, Log Analysis, Python, Regular Expression, Serangan Website*

A B S T R A C T

In the current era of digitalization, the threat of cyber attacks continues to increase globally. cyberattacks refer to criminal acts that can lead to tampering, falsification, and theft of valuable information from applications or websites. Sourced on the Suara.com page, there were more than 700 million cyber attacks that took place in Indonesia in 2022, followed by

**Penulis Korespondensi:*
email: 41519010045@student.mercubuana.ac.id

the leak of e-HAC health information. This relates to the role of the web server to serve HTTP (Hypertext Transfer Protocol) requests. In addition, the web server is tasked with translating dynamic codes into static codes on a website page. Based on this, a log analysis system was created using the boyer moore algorithm and regular expressions techniques as a search method for indications of website attacks with the result that the boyer moore algorithm and regular expressions techniques can find indications of types of attacks on websites properly and relatively quickly because it only takes 32.9 seconds to analyze 1553 rows of data coming from the log file. Thus, web server administrators can easily search for or view types of website attack attempts against web servers by cybercriminals.

Keywords: *Boyer Moore Algorithm, Flask, HTTP, Log Analysis, Python, Regular Expression, Web Application Attack*

1. PENDAHULUAN

Di era digitalisasi saat ini, ancaman serangan siber terus meningkat secara global. Serangan siber tidak lagi terbatas pada beberapa negara saja, tetapi hampir seluruh dunia merasakan dampaknya. Serangan siber merujuk pada tindakan kejahatan yang dapat menyebabkan gangguan, pemalsuan, dan pencurian informasi berharga dari aplikasi atau situs web. Serangan ini dapat berdampak serius terhadap keamanan jaringan, basis data, dan sistem komputer[1].

Bersumber pada halaman suara.com diketahui terdapat lebih dari 700 juta serangan siber yang berlangsung di Indonesia pada tahun 2022 serta baru-baru ini pula terjadi kebocoran informasi pendaftaran kartu SIM, di samping insiden- insiden besar lebih dahulu yang melibatkan informasi kesehatan e-HAC, informasi departemen, BUMN, sampai informasi klien di e-commerce terkemuka[2]. Kebocoran tersebut berkaitan dengan sisi peranan dari Web server guna melayani permintaan HTTP (*Hypertext Transfer Protocol*) dari website browser serta mengirimkan kode-kode dinamis ke server aplikasi. Server inilah yang menerjemahkan serta mengerjakan kode-kode dinamis menjadi kode-kode statis dalam suatu laman statis yang setelah itu dikirimkan ke browser oleh web server[3]. Dari kejadian tersebut terdapat dua penelitian berkaitan yang dilakukan oleh I Wayan Ardiyasa pada tahun 2020 dengan menerapkan K-Means *clustering* untuk klasifikasi serangan cyber pada syslog file berbasis website dengan menggunakan bahasa pemrograman PHP, penelitian tersebut bertujuan untuk membuat kategori tipe serangan yang dibuat oleh pelaku kejahatan siber, selain itu adapun penelitian yang dilakukan oleh Yogi, Ikhwan Ruslianto dan Syamsul Bahri pada tahun 2019 dengan menerapkan teknik regular expressions untuk mengetahui pola perilaku pengunjung website menggunakan data *log web server*.

Berdasarkan hal tersebut, penulis berniat mengembangkan dan membangun sebuah sistem detektor serangan siber berbasis website yang mengimplementasikan sistem deteksi serangan terhadap website dengan menggunakan bahasa pemrograman python flask yang disandingi dengan algoritma boyer moore dan teknik *regular expressions* yang diharapkan dapat mempermudah administrator *web server* dalam mencari potensi penyerangan keamanan website yang dilakukan oleh pelaku kejahatan siber.

2. DASAR/TINJAUAN TEORI

Dalam penelitian ini, peneliti menggunakan referensi pustaka sebagai sumber acuan dalam penelitian yang dilaksanakan dan direncanakan.

2.1. *Web Server*

Server web adalah lokasi di mana halaman web dan data terkait dengan situs web yang dibuat disimpan, sehingga pengguna dapat mengakses dan melihat data tersebut. Ada tiga jenis log yang penting untuk dipantau dalam memonitor kegiatan server web:

a. Log Akses

Log akses adalah file yang digunakan untuk mencatat semua akses yang dilakukan terhadap server web. Log ini mencatat informasi seperti alamat IP pengguna, waktu akses, halaman yang diakses, dan kode status HTTP yang dihasilkan oleh server.

b. Log Server

Log server adalah file yang mencatat kejadian-kejadian tertentu pada server web. File ini digunakan untuk memeriksa masalah jika terjadi kesalahan pada server web. Log ini dapat mencatat informasi seperti permintaan yang memakan waktu lama, kesalahan server internal, atau serangan yang mencurigakan.

c. Log Kesalahan

Log kesalahan adalah file yang mencatat setiap kesalahan yang terjadi pada server web, baik itu kesalahan pada file konfigurasi atau kesalahan dalam pembuatan situs web. Log ini berguna dalam menemukan dan memperbaiki masalah yang mungkin terjadi pada server web.

Dengan memonitor ketiga jenis log ini, administrator server web dapat melacak aktivitas pengguna, mengidentifikasi masalah, dan meningkatkan kinerja serta keamanan server web.

2.2. Algoritma Boyer Moore

Algoritma Boyer Moore adalah sebuah metode untuk mencari string dalam teks yang dikembangkan oleh R.M Boyer dan J.S Moore. Prinsip utama dari algoritma ini adalah melakukan pencarian string dengan memulai pembandingan karakter dari posisi paling kanan dalam *string* yang dicari. Algoritma ini menggunakan dua heuristik yaitu *Match Heuristic* (MH) dan *Occurrence Heuristic* (OH) untuk menghitung nilai panjang teks yang akan dilewati dalam setiap iterasi. Setelah nilai MH dan OH ditemukan, nilai-nilai tersebut dibandingkan untuk menentukan langkah loncatan yang optimal. Langkah loncatan ini digunakan untuk mempercepat proses pencarian *string* dalam teks. Dengan menggunakan algoritma Boyer Moore, proses pencarian secara umum menjadi lebih efisien dan cepat dibandingkan dengan algoritma lainnya[5].

1. Keunggulan daripada algoritma Boyer Moore

Algoritma ini melakukan perbandingan karakter dari kanan ke kiri dan menggunakan skema loncatan karakter yang besar. Hal ini memungkinkan algoritma Boyer Moore untuk mempercepat proses pencarian *string*, karena dengan melakukan sedikit perbandingan karakter, algoritma dapat langsung mengetahui apakah *string* yang dicari tidak ditemukan dan melompat ke posisi berikutnya. Dengan demikian, algoritma Boyer Moore mengoptimalkan proses pencarian dengan meminimalkan jumlah perbandingan yang perlu dilakukan.

2. Kekurangan daripada algoritma Boyer Moore

Algoritma Boyer-Moore adalah metode pencocokan pola yang bekerja dengan mencocokkan pola dari kanan ke kiri. Namun, kelemahan dari algoritma ini terletak pada kasus ketika semua karakter dalam pola memiliki kesamaan kecuali karakter terakhir atau

karakter paling kiri. Dalam situasi ini, proses pencarian akan memerlukan waktu yang relatif lebih lama.

2.3. Teknik Regular Expression

Ekspresi reguler (regexp, Regex, RE) merupakan bahasa mini guna mendeskripsikan *string* ataupun teks. Ekspresi reguler bisa digunakan untuk pencocokan *string*-ke-pola. Ekspresi reguler berbeda dari *string* normal dengan terdapatnya karakter eksklusif yang disebut metakarakter. Karakter- karakter ini tidak betul- betul sesuai dengan karakter itu sendiri, namun mewakili kumpulan karakter lain ataupun semacam pola ekslusif[4].

Ekspresi reguler kerap digunakan untuk mencocokkan serta mencari pola dalam teks, mulai dari yang sederhana sampai yang sangat kompleks. Berikut merupakan sebagian contoh karakter ekspresi reguler yang digunakan oleh Michael Fitzgerald dalam bukunya yang bertema" Pengantar Ekspresi Reguler". Pada Tabel 1 yaitu:

Tabel 1. *Pattern Regular Expression*

No	Metacharacter	Keterangan
1	/	Karakter yang digunakan sebagai pemisah dalam beberapa dialek regex, seperti dalam "/pattern/"
2	\	Karakter escape yang digunakan untuk memperlakukan karakter yang berikutnya sebagai karakter literal
3	^	Sebagai karakter pertama dalam kurung kurawal akan membalik karakter kelas, yang berarti ia akan mencocokkan karakter yang tidak ada dalam kelas tersebut. Misalnya, "[^a-z]" akan cocok dengan karakter apa pun kecuali huruf kecil
4	()	Tanda kurung digunakan untuk mengelompokkan bagian-bagian pola bersama. Misalnya, pola "(abc)+" akan cocok dengan "abc", "abcabc", "abcabcabc", dll
5	[]	Karakter tanda kurung siku digunakan untuk mencocokkan satu karakter dari sekelompok karakter yang diberikan. Misalnya, pola "[abc]" akan cocok dengan "a", "b", atau "c"
6	[<*string*>]	Regex yang mencocokkan apa pun dari karakter pada *string* dan tidak ada lainnya
7	[^<*string*>]	Regex yang mencocokkan karakter apa pun kecuali baris baru dan karakter dari *string* itu sendiri

8	+	Tanda plus menunjukkan bahwa karakter sebelumnya harus muncul satu atau lebih kali. Misalnya, pola "go+d" akan cocok dengan "god", "good", "gooood", dll., tetapi tidak akan cocok dengan "gd"
9	“	Karakter tanda kutip ganda (") digunakan untuk memperlakukan karakter sebelumnya sebagai karakter literal dalam *string*
10	.	Tanda titik mewakili satu karakter apa pun kecuali karakter baris baru. Misalnya, pola "c.t" akan cocok dengan "cat", "cut", "cot", dll
11	-	Dapat digunakan untuk menentukan kisaran karakter. Misalnya, "[a-z]" akan cocok dengan semua huruf kecil
12	$	Karakter tanda dolar ($) digunakan untuk mencocokkan akhir teks atau akhir baris. Misalnya, pola "abc$" akan cocok dengan "abc" hanya jika itu adalah akhir teks atau akhir baris
13	*	Meta karakter kuantifier yang mengindikasikan bahwa elemen sebelumnya dapat muncul nol atau lebih kali. Misalnya, "a*" akan mencocokkan "a", "aa", "aaa", dan seterusnya
14	\S	Digunakan untuk mencocokkan satu karakter yang bukan merupakan karakter spasi
15	\s	Meta karakter yang mencocokkan spasi putih
16	\d	Meta karakter yang mencocokkan angka. setara dengan karakter kelas "[0-9]", yang mencocokkan semua angka dari 0 hingga 9
17	\D	meta karakter yang mencocokkan karakter yang bukan angka. Setara dengan karakter kelas "[^0-9]", yang mencocokkan semua karakter kecuali angka 0 hingga 9

2.4. Python

Python merupakan salah satu bahasa pemrograman yang digunakan untuk membangun aplikasi, baik yang berbasis desktop, berbasis web, maupun berbasis *mobile*[6].

2.5. Flask

Flask adalah sebuah framework web yang dibangun menggunakan Python. Meskipun termasuk dalam kategori framework mikro, Flask tetap memiliki fungsionalitas yang cukup untuk mengembangkan aplikasi web dengan mudah. Meski inti dari Flask sederhana, framework ini tetap fleksibel dan dapat diperluas sesuai kebutuhan pengembang. Flask memungkinkan pengembang untuk membuat aplikasi web terstruktur dengan pengaturan tampilan yang lebih sederhana dibandingkan dengan framework lainnya[7].

3. METODE

3.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan untuk penelitian ini yaitu observasi atau studi lapangan dengan mewawancarai administrator web server untuk mengetahui letak permasalahan. Hasil wawancara tersebut diharapkan dapat memenuhi kebutuhan yang dibutuhkan oleh pengguna sehingga nantinya dapat membangun sistem dengan optimal dan efisien.

3.2 Metode Pengembangan Perangkat Lunak

Dalam pengembangan penelitian ini, metode yang digunakan adalah metode waterfall karena mengadopsi pendekatan berurutan satu per satu untuk setiap fase. Pendekatan ini dipilih karena dapat mengurangi kemungkinan kesalahan dalam pengembangan sistem[8]. Adapun tahapan pada model waterfall ini, antara lain:

a. Requirement

Pada tahap ini, dilakukan analisis kebutuhan sistem yang meliputi kebutuhan fitur pada aplikasi, database hingga data logging pada web server. Proses ini diharapkan mendapatkan informasi terkait spesifikasi yang dibutuhkan oleh pengguna aplikasi website yang dibuat. Informasi yang didapat ini bersumber dari hasil wawancara dan juga studi literatur.

b. Design

Pada tahap ini, mengimplementasikan desain aplikasi sesuai dengan kebutuhan yang sudah dianalisa seperti pembangunan struktur data, arsitektur software, perancangan interface, hingga perancangan fungsi internal dan eksternal dari setiap algoritma prosedural.

c. Implementation

Pada tahap ini, dilakukan penulisan code untuk pembuatan aplikasi web sistem deteksi serangan website ini dengan menggunakan bahasa pemrograman Python dan teknik *Regular Expression Pattern Matching* & Boyer Moore sebagai algoritma pencarian karakter.

d. Integration & Testing

Dalam tahap ini dilakukan proses integrasi dan pengujian sistem dari fitur atau modul yang sebelumnya telah dibuat sehingga nantinya dari sistem yang telah dibuat diharapkan dapat meminimalisir kesalahan seperti error atau bug dan juga memenuhi semua kriteria yang dibutuhkan.

e. Maintenance

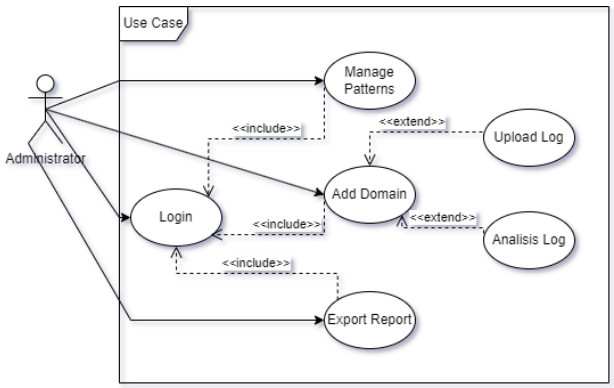
Dalam fase ini, ada kemungkinan bahwa perangkat lunak yang telah dibuat akan mengalami perubahan saat diimplementasikan oleh pengguna. Perubahan tersebut bisa

terjadi karena kesalahan yang muncul dan tidak terdeteksi selama pengujian, atau karena perangkat lunak harus disesuaikan dengan spesifikasi sistem yang baru. Oleh karena itu, diperlukan proses pemeliharaan (maintenance).

4. HASIL DAN PEMBAHASAN

4.1. *Use Case Diagram*

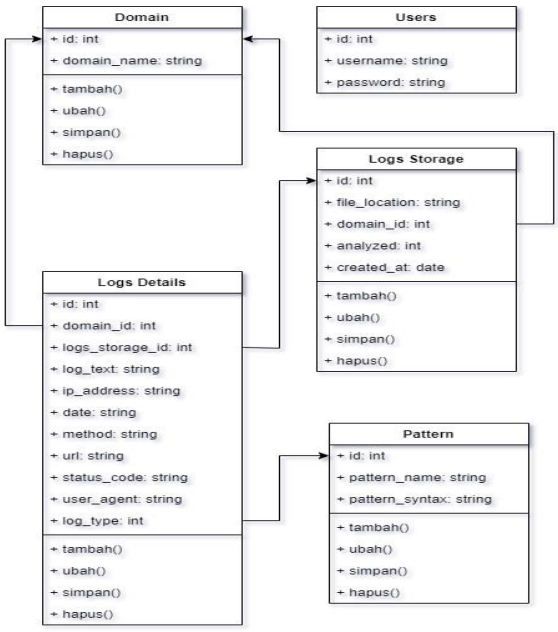
Diagram ini merupakan gambaran dari fungsi-fungsi utama yang diajukan oleh pengguna dari sisi sistem. Penggunaannya dalam kasus pembuatan sistem deteksi serangan website ini, *use case* diagram dapat menggambarkan interaksi antara pengguna (administrator) dan sistem seperti proses pengelolaan domain, pengelolaan log file, analisis log dan lain-lain.



Gambar 1. *Use Case Diagram* Aplikasi

4.2. *Class Diagram*

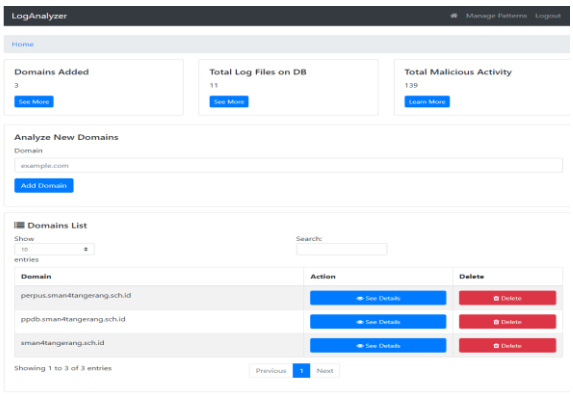
Diagram ini menggambarkan struktur data dan hubungan antar kelas dalam sistem. Dalam sistem deteksi serangan website ini, *class* diagram dapat menggambarkan kelas-kelas, atribut serta metode yang terlibat dalam sistem seperti kelas untuk domain, *log file* dan lainnya.



Gambar 2. *Class Diagram* Aplikasi

4.3. Tampilan Aplikasi

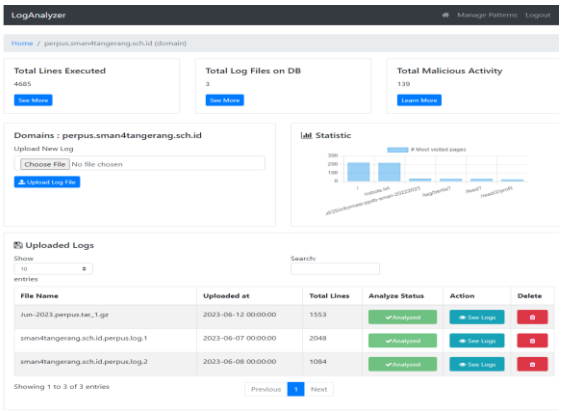
a. Halaman Dashboard



Gambar 3. Halaman Dashboard

Pada halaman ini, di tampilkan seluruh informasi termasuk domain, total log files, dan total aktifitas upaya penyerangan yang terdeteksi oleh sistem.

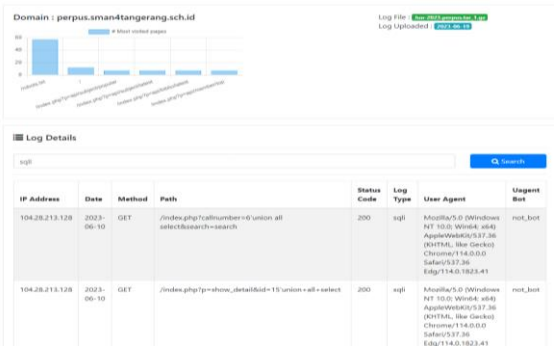
b. Halaman Detail Domain



Gambar 4. Halaman Detail Domain

Pada halaman ini, di tampilkan statistik halaman website yang paling sering di kunjungi, *log file* yang sudah di unggah, dan total serangan terhadap domain tersebut.

c. Halaman Detail *Log*



Gambar 5. Halaman Detail *Log*

Pada halaman ini, di tampilkan statistik halaman website yang paling sering di kunjungi, tanggal log file di unggah, dan hasil analisa serangan dalam log tersebut. Hasil analisa dapat di lihat di dalam kolom *log type* dan *uagent bot.*

d. Hasil Analisa

Gambar 6. Hasil Analisa Eksekusi 1553 Baris *Log*

Dalam **Gambar 6** dihasilkan waktu eksekusi selama 32.9 detik untuk proses analisa 1553 baris *log* yang tergolong cepat dengan bantuan *library concurrent* pada pemrograman python agar proses tersebut berjalan secara paralel sehingga dapat mempercepat waktu analisa.

4.4. Hasil Pengujian

a. Algoritma Boyer Moore

Pada pengujian algoritma boyer moore ini, dilakukan pencocokan kata pada teks user-agent yang terindikasi sebuah *crawler bot* atau bahkan *hacking bot*. dalam pencocokan ini dilakukan pencarian karakter pattern pada teks.

Tabel 2. Nilai OH dan MH pada pattern “sqlmap”

Indeks	**0**	**1**	**2**	**3**	**4**	**5**
Pattern	s	q	l	m	a	p
OH	5	4	3	2	1	0
MH	6	6	6	6	6	1

Tabel di atas merupakan perhitungan dari pencarian OH yaitu: “panjang teks (length) – 1 – Indeks”, berikut jabaran dari perhitungan OH:

Tabel 3. Jabaran Perhitungan Nilai OH

s	=	K1	=	6	-	1	-	0	=	5
q	=	K2	=	6	-	1	-	1	=	4
l	=	K3	=	6	-	1	-	2	=	3
m	=	K4	=	6	-	1	-	3	=	2
a	=	K5	=	6	-	1	-	4	=	1
p	=	K6	=	6	-	1	-	5	=	0

Untuk pencarian MH yaitu nilainya sama dengan jumlah pattern yaitu 6, namun jika nilai OH bernilai 0 maka nilai MH-nya 1.

1. Tahap Pertama
   Pada tahap pertama, pencocokan string pattern dimulai dengan deretan string teks yang pertama “sqlmap”

Posisi Teks	0	1	2	3	4	5	6	7	8	9	10	11	12
Teks	:	/	/	s	q	l	m	a	p	.	o	r	g
Pattern	s	q	l	m	a	p							

   Pada tahap ini proses pencocokan di mulai dari kanan yaitu karakter “p”, namun pada karakter “p” pada pattern tidak memiliki kecocokan dengan karakter “l”, sehingga akan bergeser nilainya berdasarkan tabel *Occurence Heuristic* (OH) dan *Match Heuristic* (MH) pada tabel 4.16. Pada tabel OH dan MH, karakter “l” ditemukan maka pergeseran dilakukan sebanyak “3” langkah namun jika tidak ditemukan maka pergeseran dilakukan sebanyak 6 langkah.

2. Tahap Kedua

Posisi Teks	0	1	2	3	4	5	6	7	8	9	10	11	12
Teks	:	/	/	s	q	l	m	a	p	.	o	r	g
Pattern				s	q	l	m	a	p				

   Dalam tahap sebelumnya karakter “l” merupakan karakter yang termasuk dalam tabel OH dan MH yang berarti 3 langkah pergeseran dilakukan berdasarkan perhitungannya. Maka karakter “l” di sejajarkan dengan “l” yang berada pada teks.

3. Tahap Ketiga

Posisi Teks	0	1	2	3	4	5	6	7	8	9	10	11	12
Teks	:	/	/	s	q	l	m	a	p	.	o	r	g
Pattern				s	q	l	m	a	p				

   Dalam tahap ini, pencocokan dilakukan kembali secara berurutan dari arah kanan ke kiri, apakah karakter “p” pada pattern sudah sama atau belum dengan “p” pada teks. Hal tersebut di cocokkan secara berulang hingga karakter “s” pada teks dan pattern sesuai. Pada tahap ini didapatkan kesamaan dari semua karakter yang ada pada pattern sehingga dapat dihasilkan kata kunci “sqlmap” ditemukan pada teks “://sqlmap.org”.

IP Address	Date	Method	Path	Status Code	Log Type	User Agent	Uagent Bot
104.28.213.128	2023-06-10	GET	/index.php?p=show_detail&id=15'union+all+select	200	sqli	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.41	not_bot
8.219.247.104	2023-06-10	GET	/index.php?p=show_detail&id=15&EGCJ=9206 AND 1=1 UNION ALL SELECT 1,NULL,'<script>alert("XSS")</script>',table_name FROM information_schema.tables WHERE 2>1--/**/; EXEC xp_cmdshell('cat ../../../etc/passwd')#	403	sqli	sqlmap/1.2.4#stable (http://sqlmap.org)	sqlmap
180.252.253.0	2023-06-10	GET	/index.php?p=show_detail&qREB=2064 AND 1=1 UNION ALL SELECT 1,NULL,'<script>alert("XSS")</script>',table_name FROM information_schema.tables WHERE 2>1--/**/; EXEC xp_cmdshell('cat ../../../etc/passwd')#	403	sqli	sqlmap/1.7.2#stable (https://sqlmap.org)	sqlmap

Gambar 7. Hasil Pengujian Boyer Moore

```
Text : Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/114.0.5735.90 Mobile Safari/537.36 (compatible; Googlebot/2.1; +http://www.google
.com/bot.html) has been found in googlebot
Text : Mozilla/5.0 AppleWebKit/537.36 (KHTML, like Gecko; compatible; bingbot/2.0; +http://www.b
ing.com/bingbot.htm) Chrome/103.0.5060.134 Safari/537.36 has been found in bingbot.htm
Text : Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/114.0.5735.90 Mobile Safari/537.36 (compatible; Googlebot/2.1; +http://www.google
.com/bot.html) has been found in googlebot
```

Gambar 8. Hasil Pengujian Algoritma Boyer Moore (Proses *Backend*)

Dari hasil pengujian algoritma di atas didapatkan *crawler bot* bernama “sqlmap” yang di gunakan untuk upaya penyerangan terhadap website.

b. Teknik *Regular Expression*

Pada pengujian ini, dilakukan pengujian dari implementasi regex yang di gunakan untuk memilah informasi yang di perlukan dari isi file log pada tiap-tiap baris dan juga proses deteksi serangan website dari sisi *pattern* yang biasa digunakan dalam *hacking* menggunakan metode “SQL *Injection*, XSS dan LFI”. Berikut merupakan proses deteksi serangan dengan mengambil salah satu contoh upaya serangan SQL *Injection* yang diambil dari *access log* website perpus.sman4tangerang.sch.id menggunakan *pattern* yang sudah di input sebelumnya dalam database.

Pattern:

SQLi	\b(select|insert|update|delete|drop|create|alter)\b

Penjelasan dari pattern tersebut adalah sebagai berikut:

1. **\b**: Ini menandakan batas kata (word boundary). Memastikan bahwa kata yang cocok harus berada pada batas kata penuh.
2. **(select|insert|update|delete|drop|create|alter)**: Ini adalah grup yang mencocokkan dengan salah satu kata kunci SQL yang umum digunakan dalam serangan SQL Injection, seperti *"select", "insert", "update", "delete", "drop", "create", atau "alter"*.
3. **\b**: Ini kembali menandakan batas kata (*word boundary*), memastikan kata kunci SQL cocok secara penuh.

104.28.213.128	2023-06-10	GET	/index.php?p=show_detail&id=15'union+all+select	200	sqli	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.41	not_bot

Gambar 9. Hasil dari *Pattern* SQL *Injection*

Dari Gambar 9 didapatkan hasil yaitu upaya serangan SQL *Injection* yang dilakukan oleh IP address 104.28.213.128 tanpa menggunakan *bot* atau manual.

5. KESIMPULAN

Berlandaskan hasil perancangan, implementasi serta pengujian yang dicoba maka diambil kesimpulan sebagai berikut:

1. *Log file* yang sudah di unggah akan di analisa oleh sistem dimulai dari halaman yang berisi tentang halaman yang paling banyak di kunjungi untuk mengetahui halaman yang paling sering di akses oleh pengunjung, jumlah upaya serangan yang di lakukan oleh penyerang, tipe upaya serangan yang dilakukan oleh penyerang dan file berupa *report* serangan dari tiap-tiap *log file* yang di unggah.
2. Didapatkan hasil bahwa algoritma boyer moore dapat mendeteksi *crawler bot* dengan baik berdasarkan daftar user-agent terindikasi bot.
3. Didapatkan hasil bahwa teknik *regular expression* dapat di implementasikan dengan baik sehingga dapat mendeteksi tipe upaya serangan yang dilakukan oleh pelaku serangan.
4. Didapatkan hasil bahwa sistem ini membutuhkan waktu hanya 32.9 detik untuk menganalisis 1553 baris dari isi file log yang tergolong cepat karena menggunakan proses paralel dari *library concurrent* pada python.

REFERENSI

[1] S. Parulian, D. A. Pratiwi, dan M. Cahya Yustina, “Ancaman dan Solusi Serangan Siber di Indonesia.” [Daring]. Tersedia pada: http://ejournal.upi.edu/index.php/TELNECT/

[2] D. Novianty, “Marak Kebocoran Data, Survei: Indonesia Kekurangan Tenaga Ahli Keamanan Siber.” https://www.suara.com/tekno/2022/10/02/090719/marak-kebocoran-data-survei-indonesia-kekurangan-tenaga-ahli-keamanan-siber (diakses 2 Oktober 2022).

[3] I. Gusti, L. Putra, dan E. Prismana, “Implementasi Load Balancing Pada Web Server Dengan Menggunakan Apache Implementasi Load Balancing Pada Web Server Dengan Menggunakan Apache Supramana.”

[4] I. Ruslianto, S. Bahri, J. Rekayasa Sistem Komputer, dan J. H. Hadari Nawawi, “Analisa Log Web Server Untuk Mengetahui Pola Perilaku Pengunjung Website Menggunakan Teknik Regular Expressions,” 2019. [Daring]. Tersedia pada: https://httpd.apache.org/docs/2.4/logs.HTM

[5] S. R. Siregar, “Penerapan Algoritma Boyer Moore Pada Aplikasi Kumpulan Cerita Motivasi.”

[6] A. A. Hasanuddin, “Rancang Bangun Web-GIS Berbasis Geodjango-Python.”

[7] R. Irsyad, “Penggunaan Flask untuk Pemula”.

[8] W. Gunawan dan B. S. DP, “JEPIN (Jurnal Edukasi dan Penelitian Informatika),” *Jurnal Edukasi dan Penelitian Informatika*, vol. 6, no. 2, 2020.