Digital Transformation Technology (Digitech) | e-ISSN : 2807-9000
Volume 3, **Number** 1, Maret 2023
https://doi.org/10.47709/digitech.v3i1.2764

Mendesain Cyber Security Untuk Mencegah Serangan DDoS Pada Website Menggunakan Metode Captcha

Jerry Hansen[1*], Tata Sutabri[2]
[1,2,]Universitas Bina Darma Palembang, Indonesia,
[1]jerryxhansen@gmail.com, [2]tata.sutabri@gmail.com

Histori Artikel:
Diajukan: 24 Agustus 2023
Disetujui: 25 Agustus 2023
Dipublikasi: 26 Agustus 2023

Kata Kunci:
Cyber; Security; DDoS; Website; Captcha

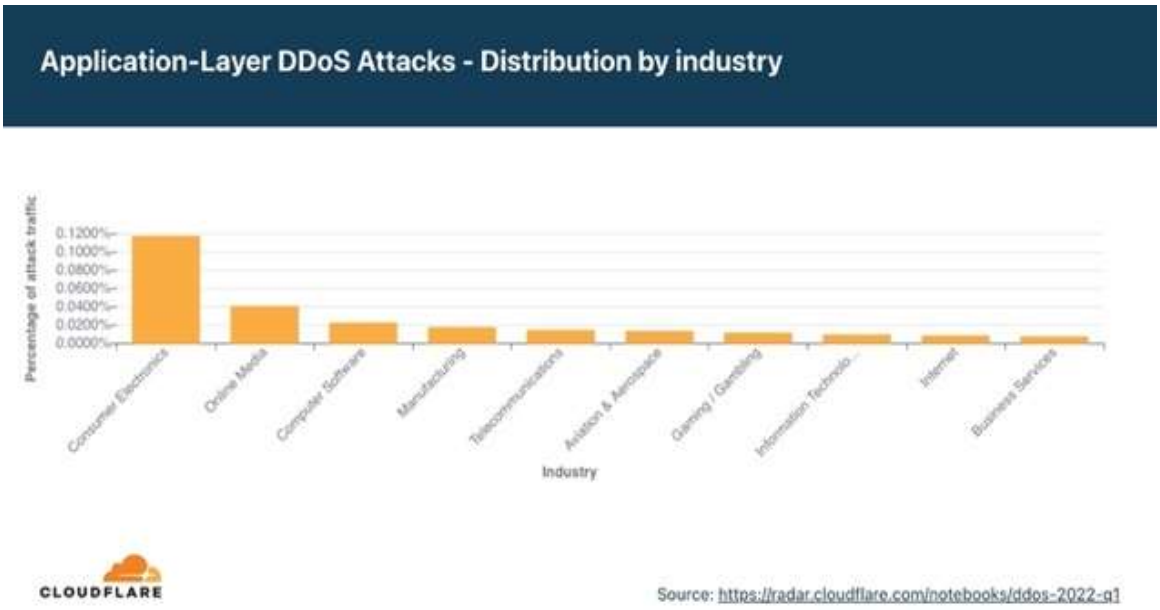
Abstrak
Serangan Distributed Denial of Service (DDoS) dari tahun ke tahun menjadi ancaman serius bagi infrastruktur jaringan dan sistem dalam lingkungan teknologi informasi. Hal ini harus menjadi perhatian bagi semua pihak yang memanfaatkan teknologi informasi, karena dampak yang akan ditimbulkan terhadap serangan ini sangat merugikan. Pada penelitian ini kami mengkaji dan mendesain penerapan metode captcha sebagai pendekatan dalam keamanan siber untuk melindungi sistem dari serangan DDoS. Captcha telah digunakan secara luas untuk membedakan antara manusia atau bukan, dan implementasinya dalam pertahanan DDoS akan dibahas dan dijelaskan dalam penelitian ini. Desain yang kami gunakan, selanjutnya kami implementasikan ke dalam baris kode yang berbasis website sebagai sampel dari penerapan metode captcha yang dipakai. Dalam penelitian ini, kami menggunakan metode pendekatan kualitatif. Data yang digunakan menggunakan analisis peneliti. Pengujian uji coba metode ini dilakukan dengan menggunakan penggunaan captcha secara efisien menggunakan perangkat pengembangan berbasis website sebagai sampel untuk dilakukan analisis terhadap kekuatan yang dihadirkan dari penggunaan metode captcha. Hasil penetian ini menunjukkan efektivitas captcha dalam mengurangi dampak serangan DDoS.

PENDAHULUAN

Tren pengembangan maupun penggunaan teknologi terus meningkat seiring kemajuan zaman. Informasi konvensional yang dulu bisa kita dapatkan dengan media cetak, beralih ke dalam teknologi yang membuat informasi dan transaksi yang disajikan secara cepat dan hangat karena dilakukan secara digital (Sutabri, 2012). Fenomena ini memberikan gambaran bahwa perubahan dan kemajuan teknologi berdampak kepada kemudahan bagi kehidupan manusia (Pratama & Sutabri, 2023). Penggunaan teknologi dan tersedianya transaksi yang diproses digital melalui komputerisasi menjadikan setiap sektor berlomba menggunakan teknologi informasi (Sutabri, 2012). Namun, dibalik fenomena peralihan penggunaan teknologi komputerisasi tersebut terdapat juga faktor negative yang mengarah pada kejahatan dalam teknologi, misalnya pencurian data, atau serangan DDoS yang perlu menjadi perhatian sebab akan berpengaruh pada teknologi itu sendiri. Faktor yang menyangkut keamanan pada teknologi informasi haruslah diperhatikan. Karena hal ini terkait dengan integritas dan kelanjutan dan kerahasiaan, dan reputasi pengguna teknologi itu sendiri (Sutabri, 2012).

Jika kita melihat data, secara global hampir semua bidang dan sektor industri juga tak luput dari serangan DDoS ini. Industri *Consumer Electronics* menjadi yang paling pertama banyak diserang dengan peningkatan sebesar 5.086% QoQ. Diikuti oleh industri Media Online dengan peningkatan serangan QoQ sebesar 2.131%. Selanjutnya adalah perusahaan *Software* Komputer, dengan peningkatan sebesar 76% QoQ dan 1.472 YoY.tersebut. (Yoachimik, 2022). Data *Application-Layer DDoS Attacks-Distribution by Industry* tersebut bisa kita lihat pada Gambar 1 seperti berikut.

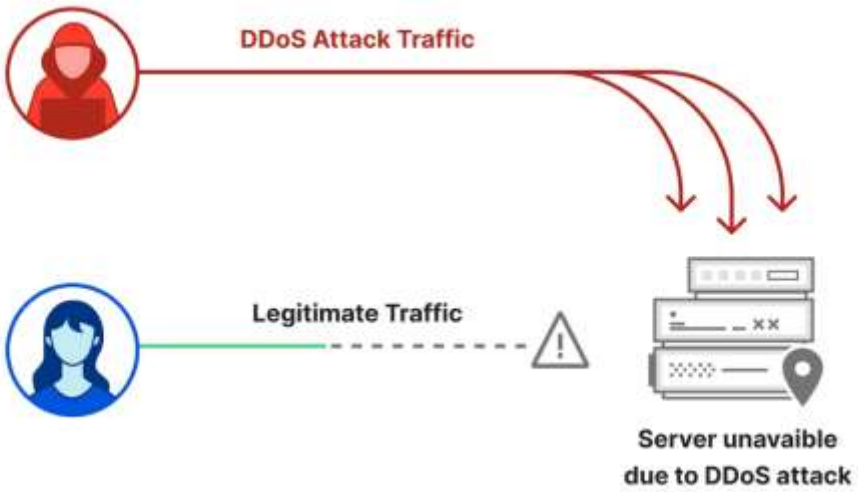
https://doi.org/10.47709/digitech.v3i1.2764



Gambar 1. *Application-Layer DDoS Attacks-Distribution by Industry*

Sumber: (Yoachimik, 2022)

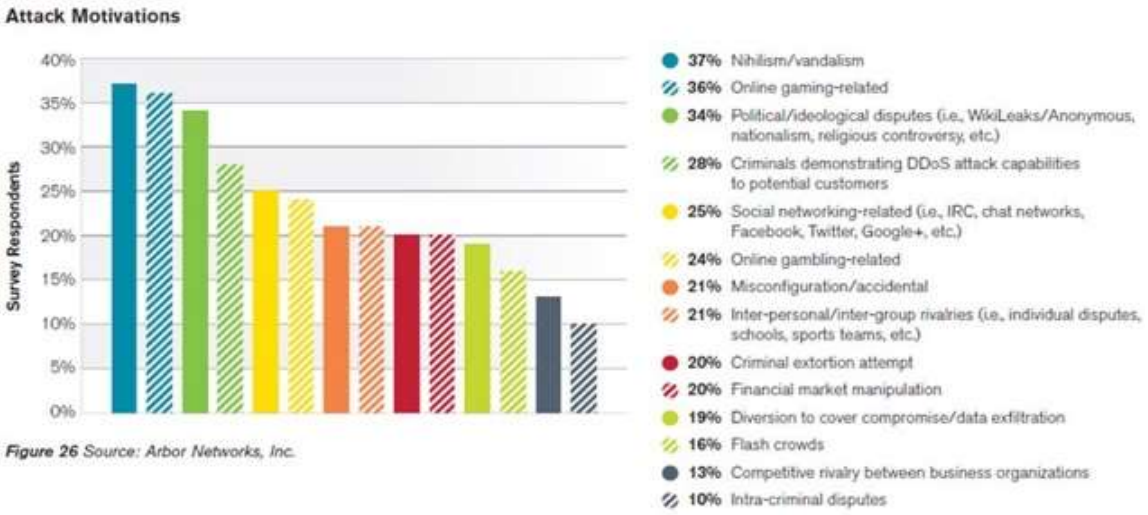
Setiap organisasi, industri, bahkan individu dapat menjadi target serangan DDoS. Serangan DDoS ini sering dilakukan dengan maksud untuk menolak pengguna yang sah untuk mengakses layanan. Membuat dan mengganggu pusat *server website* dengan membuatnya tidak dapat memproses permintaan pengguna yang sah. Jika pusat *server website* dibombardir dengan lebih banyak permintaan dari jumlah yang dapat diprosesnya, pusat *server website* akan membatalkan permintaan yang sah, yang mengakibatkan penurunan kinerja dan hingga menutup akses untuk pengguna yang sah. Pada Gambar 2 kita dapat melihat ilustrasi skema serangan DDoS.



Gambar 2. Skema Serangan DDoS
Sumber: (Masolo, 2023)

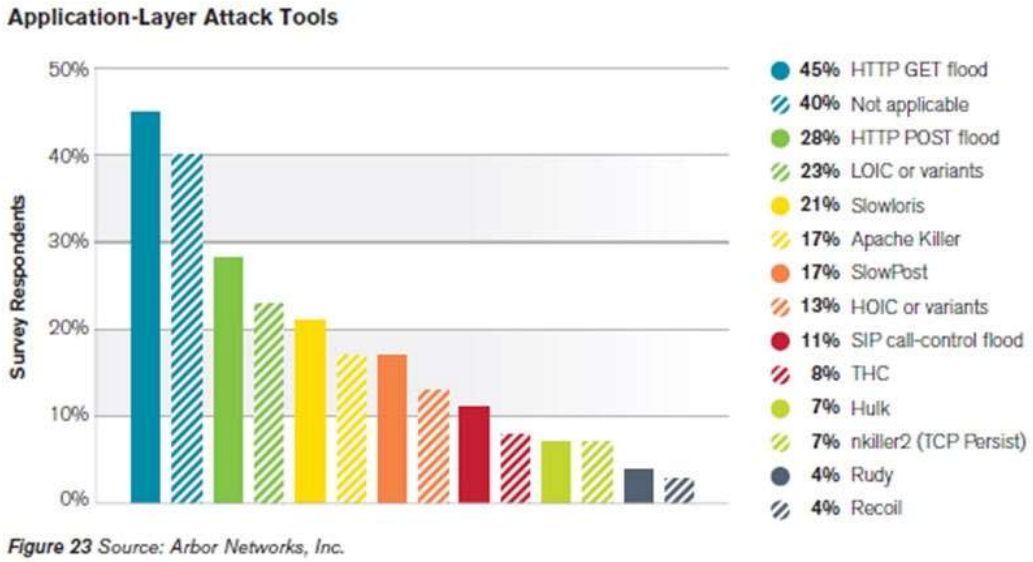
Selain itu, karena tipe serangan ini memungkinkan mereka mengakses semua sumber daya yang dimiliki korbannya, pelaku dari serangan DDoS tersebut biasanya menjadikan DDoS sebagai bagian dari tindak kejahatan lainnya misalnya ditujukan untuk memeras uang dari target korbannya atau mengganggu operasi jalannya organisasi. Karena itulah, pemahaman mengenai DDoS sangat penting untuk menetapkan metode yang efektif untuk mengurangi kerusakan akibat serangan ini (Security, 2022). Motivasi serangan DDoS dapat dilihat pada

Gambar 3 berikut.



Gambar 3. *Attack Motivation*
Sumber: (Security, 2022)

Berdasarkan data dari Arbor Network, Inc yang bisa dilihat pada Gambar 4, serangan lapisan aplikasi terdiri dari sekitar 17% dari semua serangan DDoS yang dilaporkan. Mereka menargetkan paket aplikasi web untuk mengganggu transmisi data antar host. Sebagai contoh, *HTTP Get flood* yang menggunakan beberapa mesin yang terinfeksi untuk memaksa target mengeluarkan sumber daya dalam jumlah berlebihan saat merespons permintaan HTTP. Dari kacamata penyerang, HTTP *Get flood* adalah ancaman yang jauh lebih efektif daripada jenis serangan lainnya karena tidak perlu menghabiskan banyak bandwidth untuk mengunci server. Meskipun HTTP *Get flood* biasanya merupakan serangan lapisan aplikasi yang paling umum dialami, namun hal ini hanyalah salah satu dari banyak alat serangan lapisan aplikasi yang tersedia (Security, 2022).

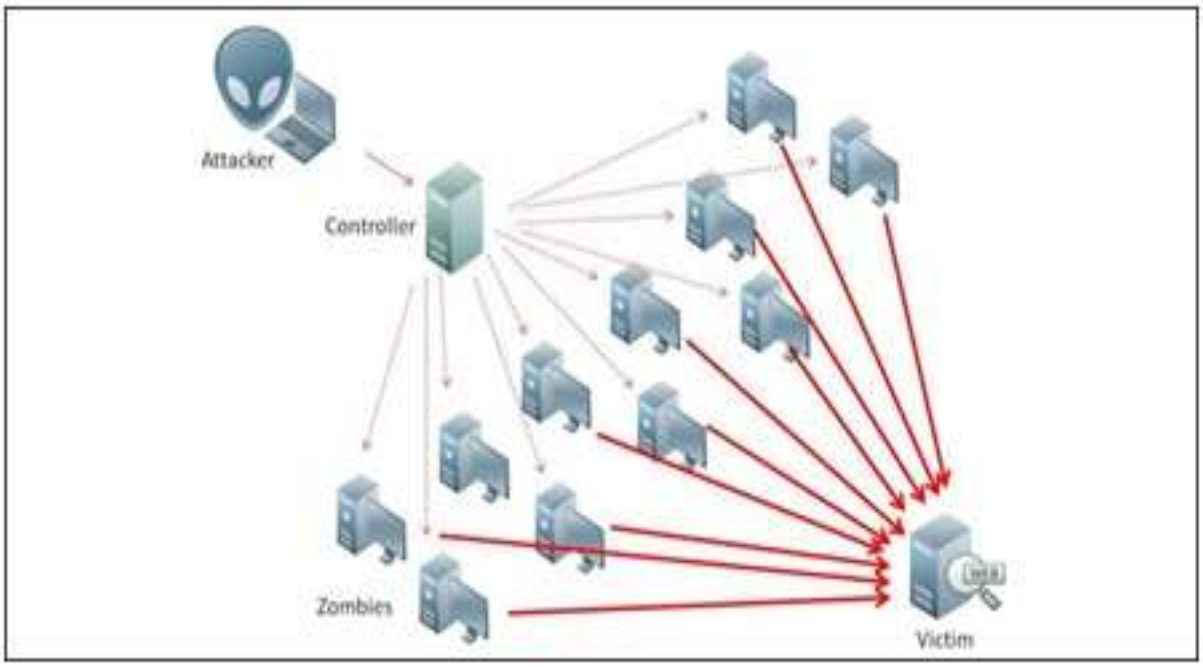


Gambar 4. *Application-Layer Attack Tools*
Sumber: (Security, 2022)

DDoS *(Distributed Denial of Service)* adalah jenis serangan yang digunakan untuk mengganggu hak akses pengguna yang dilakukan secara massif (Juniska, et al., 2022). Dimana penyerang menggunakan banyak perangkat yang terdistribusi untuk membanjiri sistem target dengan lalu lintas yang berlebihan. Tujuannya adalah untuk menghabiskan sumber daya sehingga layanan menjadi tidak dapat diakses (Kusuma & Artha, 2022). Alur

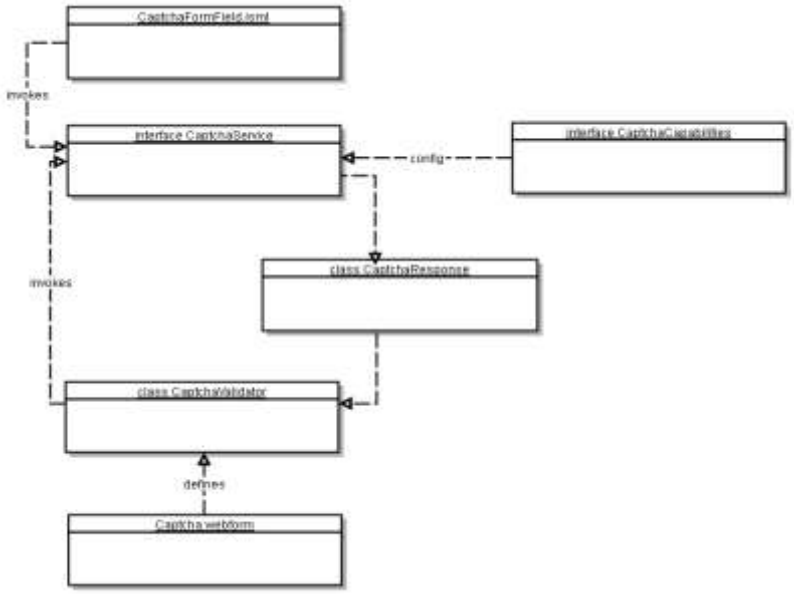
serangan DDoS ini dikerjakan oleh penyerang melalui cara pengiriman layanan permintaan yang palsu atau melalui permintaan secara terus menerus sehingga sistem gagal melayani permintaan lain atau bahkan down, *error*, dan *blank* yang dilakukan secara terdistribusi. Yakni penyerang dapat merusak beberapa mesin, kemudian mesin tersebut menjadi berubah dan mengendalikan beberapa mesin lagi lainnya, hingga akhirnya penyerang akan mengendalikan mesin-mesin tersebut secara terdistribusi untuk menyerang korban dan untuk menghilangkan informasi dari korbannya (Hermawan, 2016).

Serangan ini adalah salah satu serangan yang sering dijumpai. Serangan DDOS dapat terjadi pada kapan saja dan ke siapa saja dalam sebuah jaringan komputer, bahkan ke pengguna pribadi. Tetapi yang sering ditemukan, serangan ini menyerang server yang memberikan layanan dan kepada banyak pengguna misalnya perusahaan internet, sosial, serta perbankan yang memberikan pelayanannya berjalan pada jaringan. Bertujuan agar supaya pelayanan dapat dimatikan dari komputer atau jaringan yang menjadi tujuan serangan. Dampaknya akan sangat merugikan bagi perusahaan, organisasi ataupun isntansi yang menyediakan pelayanan terhdap jasa para penggunanya terutama bagi perbankan, karena dapat mengaggu jalannya hingga mematikan pelayanan pada sistem. Ini akan membuat pengguna yang absah tidak menerima dan memperoleh pelayanan. Seperti yang terlihat pada Gambar 5, pada prinsipnya serangan DDOS ini sulit untuk dideteksi, kecuali hanya jika sudah melakukan beberapa kali percobaan oleh penyerang dengan idetifikasi awal memastikan penyerang identik terhadap alamat IP sama. Masalah lainnya, DDOS relatif sulit untuk dicegah karena serangan ini pada juga memiliki keterkaitan dengan pelayanan yang ada dan diberikan dari sebuah perusahaan atau organisasi. Menariknya, sistem dengan tingkat keamanan yang tinggi umumnya akan memberikan interaksi kenyamanan yang rendah bagi para penggunanya (Hermawan, 2016).



Gambar 5. *DDoS Attacks Illustration*
Sumber: (Stewart, 2016)

Terkait serangan DDoS pada saat ini memicu penggunaan teknik atau metode yang digunakan untuk melindungi aset digital. Salah satu yang cukup banyak digunakan adalah *captcha*. Captcha digunakan untuk menghindari bot memanipulasi layanan web. Era saat ini, berbagai jenis captcha dihadirkan, terutama untuk meningkatkan keamanan dan kegunaan terhadap robot baru dan penjahat dunia maya yang melakukan tindakan destruktif. *Captcha* juga merupakan bagian dari konsep kriptografi (Trong, et al., 2023). Konsep *captcha* dapat dilihat pada Gambar 6 berikut.



Gambar 6. *Captcha Framework*
Sumber: (Intershop, 2023)

Dalam penerapannya, *captcha* biasanya diterapkan pada halaman web, login, atau area akses terbatas lainnya. Pengguna harus menyelesaikan tes *captcha* sebelum mereka diizinkan untuk mengirimkan data atau mendapatkan akses ke fitur tertentu. *captcha* dapat berbentuk gambar dengan teks yang sulit dibaca, teka-teki matematika sederhana, *re-captcha* yang meminta pengguna untuk memverifikasi gambar, dan banyak varian lainnya.

Penelitian ini dilakakukan untuk mengkaji kekuatan *captcha* sebagai salah satu upaya kemanan dari serangan DDoS yang akan dilakukan dengan uji coba sampel dari penggunaan *captcha* pada perangkat pengembangan berbasis website. Hasil penelitian ini bertujuan untuk memberikan gambaran tentang kekuatan metode *captcha*. Hasil dari ujicoba ini diharapkan dapat memberi tambahan pemahaman yang lebih mengambbarkan captcha dalam melindungi sebuah sistem teknologi informasi dalam hal ini berbasis pengembangan website.

STUDI LITERATUR

Jurnal penelitian lainnya yang terkait penggunaan *captcha* pada penelitian ini adalah penelitian yang dilakukan Prasetyo, dkk (2015), dalam penlitian yang berjudul Keamana. Authentikasi Hotspot Menggunakan Captcha, menjelaskan bahwa penggunaan alat miktrotik tidak bisa memberikan keamanan jaringan sistem hostpot. Kode *captcha* yang dilakukan pada proses autentikasi dimunculkan dengan acak, hal ini membuat kode akan selalu berubah untuk melakukan proses autentikasi Ketika halaman dilakukan penyegaran. Adanya tambahan authentikasi ini membuat sebuah kemudahan untuk pengguna dalam berinteraksi dengan hotspot.

Penelitian berikutnya yang dilakukan oleh Hermawan (2016) yang beertopik Analisis Konsep dan Cara Kerja Seranagan Komputer *Distributed Denial of Service (DDoS)* menjelaskan tentang bagaimana serangan terhadap layanan internet yang sering mengalami serangan oleh *hacker*. Serangan yang sering terjadi tersebut berjenis serangan DDoS, menyasar pada gangguan pada kinerja sebuah layanan. Damapak yang dihasilkan menyebabkan sumber daya yang dipakai akan habis dan berakibat pada kerusakan pada layanan. Untuk itu perlu adanya tambahan metode dalam mencegah serangan ini.

Selanjutnya penelitian yang dilakukan Kusuma (2022), menerangkan dalam penelitiannya yang berjudul Sistem *Firewal* Untuk Pencegahan DDoS Attack di Masa Pandemi Covid-19, diketahui bahwa sistem berbasis website saat masa pendemi mengalami peningkatan akses sehingga diperlukan upaya perlindungan untuk mencegah serangan DDoS. Hal ini dapat dilakukan dengan memberikan metode autentikasi menggunakan enkripsi, dekripsi, ataupun *captcha,* sehingga website tidak bisa diserang secara langsung.

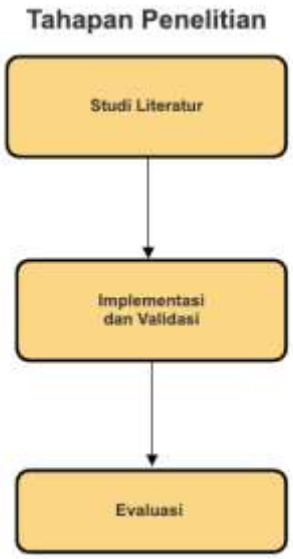
Sedangkan penelitian yang dilakukan Radiyah (2017) yang berjudul Tingkat Kenyamanan Pengguna Captcha Menggunakan Aplikasi Berbasis Web memberikan informasi bahwa diketahi kenyamanan penggunan terhadap penggunaan *captcha* dapat dilakukan dengan berbagai jenis dan kombinasi *captcha* itu sendiri tidak mengganggu kenyamanan pengguna. Namun pengguna lebih banyak memilih kombinasi *captcha* yang berbasis teks.

Penelitian yang dilakukan Safitri, dkk (2020) dengan judul Analisis Keamanan Sistem Informasi E-Banking di Era Digital Industri 4.0: Studi Literatur juga memberikan penjelasan tentang *captcha* yang ditujukan untuk menangkal serang otomatis pada halaman website. Hal ini dikarenakan akan sulit pengguna yang tidak sah untuk mengenali *captcha* yang akan memproses data dari sebuah website.

Dalam penelitian yang berjudul Meningkatkan Keamanan Terhadap SQL-Injection Studi Kasus Sistem Kepegawaian BNN yang ditlakukan oleh Andarini, dkk (2023) dijelaskan bahwa menggunakan langkah-langkah keamanan sistem misalnya SSL dan *captcha* dapat mencegah terjadinya serangan SQL *Injection*. Sehingga memberikan keamanan yang maksimal pada sistem berbasis website.

METODE

Penelitian ini menggunakan pendekatan penelitian kualitatif karena yang berkaitan dengan data yang dipakai menggunakan analisa peneliti. Pada praktiknya peneliti akan menganalisis kekuatan dan potensi lain yang dihadirkan oleh metode *captcha*. Peneliti mengumpulkan data terlebih dahulu melalui literatur, kemudian menyiapkan keranka sampel kode pengembangan berbasis website untuk melakukan implementasi dan validasi penggunaan metode *captcha*, dan pada akhirnya peneliti melakukan evaluasi hasil implementasi yang telah dilakukan. Secara lengkap, Gambar 7 berikut adalah tahapan penelitian yang dilakukan:



Gambar 7. Tahapan Penelitian

Peneliti memulai dengan studi literatur dengan mengumpulkan informasi berupa data yang tersedia baik melalui buku ataupun internet. Setelah dikumpulkan data tersebut lalu diolah dan disiapkan untuk divalidasi dengan tujuan data tersebut dapat diimplementasikan kedalam penelitian. Peneliti menggunakan metode *captcha* karena *captcha* banyak dipakai dalam membedakan antara manusia dan bukan khususnya dalam pengembangan sistem berbasis website, karena pada kenyatannya bahwa serangan DDoS ini banyaknya diakibatkan lalu lintas yang didesain untuk bekerja secara otomatis menggunakan perintah tertentu.

Lebih rinci *captcha* atau *Completely Automated Public Turing test to tell Computers and Humans Apart* atau yang lebih dikenal dengan CAPTCHA adalah sebuah metode untuk membedakan penggunan antara pengguna manusia atau bukan (Prasetyo, et al., 2015). Bertujuan untuk mengidentifikasi dan memisahkan antara akses yang valid oleh pengguna manusia dan dengan akses yang tidak valid dari apapun yang bukan pengguna manusia misalnya robot atau perintah *script* (Radiyah, 2017). Ini membantu mencegah akses otomatis yang digunakan dalam serangan seperti serangan DDoS, dan pencurian data.

Selanjutnya dalam implementasi dan validasi bahwa penggunaan *captcha* untuk melindungi dan memperpertahankan dari serangan DDoS adalah langkah dan skenario yang dapat diguakan adalah sebagai berikut yakni:

- Integrasi *captcha* pada sistem web

Pengembang dapat memasukkan tes *captcha* ke dalam pengengembangan sietem web mereka pada formulir atau area yang memerlukan akses dari pengguna, seperti formulir pendaftaran atau login atau verifikasi.

Pengguna harus menyelesaikan tes *captcha* sebelum data mereka diterima atau akses diberikan.

- Memilih tipe *captcha*

Memilih jenis ca*ptcha* yang sesuai dengan kebutuhan sistem yang digunakan dan memilih tingkat keamanan yang diinginkan. Pada beberapa jenis *captcha* bisa terdiri dan mencakup gambar, teks, atau kombinasi dari gambar, teks, angka, menyusun pola atau bahkan interaksi dengan elemen halaman.

- Mengatur tingkat kesulitan

Kita dapat mengatur tungkat kesulitan *captcha* agar menantang dan menyulitkan akses dari yang bukan semestinya. Namun perlu juga menyesuaikan agar penggunaan *captcha* tidak menyulitkan interaksi dan mengganggu pengalaman dari *user* atau pengguna.

- Menambah Proteksi Layer

Captcha harus diterapkan sebagai salah satu dari beberapa lapisan perlindungan dalam sistem pertahanan DDoS. Jika serangan DDoS terdeteksi, sesuaikan tingkat kesulitan *captcha* sesuai kebutuhan. Misalnya dengan membuat *captcha* menjadi fleksibel dan dinamis jika terdeteksi anomaly aneh saat website memantau perilaku yang aneh atau tidak biasa.

Tahapan terakhir yakni evaluasi. Setelah implementasi dan validasi dilakukan, selanjutnya peneliti melakukan evaluasi penggunaan metode *captcha* terhadap penggunaannya pada pngembangan berbasis website. Peneliti mengevaluasi dengan menarik kesimpulan bahwa setelah menerapkan dan memahami cara keraja dan penggunaan *captcha,* bahwa *captcha* dapat menjadi pilihan saat kita mengembangkan sebuah sistem bebasis website. Karena pengguna tidak dapat masuk lebih jauh, sebelum *captcha* yang dimasukkan benar dan identik dengan *captcha* yang di *generate.* Hal ini akan membantu mencegah lalu lintas dari sebuah pola yang aneh, sebab *captcha* dapat melakukan proteksi tambahan jika mendeteksi pergerakan yang tidak biasa. Namun ada baiknya pengguaan *captcha* juga dibuat dan dilakukan dengan berbagai variasi kominasi dan rumus pada saat di *generate.* Serta fleksibel dan dinamis apabila mendeteksi perilaku tida biasa saat sistem pengembangan berbasis website diakases oleh pengguna yang tidak teridentifikasi sebagai *human* atau manusia.

HASIL

Setelah melakukan pengumpulan data melalui literatur tentang teknik *captcha,* memahami pola, prinsip pada *captcha* yang akan membantu dalam analisa dalam menerapkan *captcha* pada pengembangan sistem berbasis web, lalu dilanjutkan dengan penerapan dan validasi metode *captcha* tersebut kepada sampel sistem yang digunakan pada pengembangan web. Ini dilakukan agar dapat memastikan bahwa metode *captcha* dapat bekerja dan berfungsi hingga dapat dilakukan analisa dan validasi terhadap penggunaanya. Hal ini dilakukan agar kita dapat mengetahui bagian apa saja yang perlu disesuaikan dalm menerapkan *captcha*. Peneliti memulai pengembangan kode berbasis website dengan sampel kode dengan menggunakan HTML *(Hyper Text Markup Language), javascript,* dan CSS (Cascading Style Sheets).

Diawali dengan membuat kode berbasis *website* untuk formulir dengan memasukan jalur agar fungsi *captcha* pada kode *javascript,* dapat diimplementasikan. Formulir dibuat untuk membuat pengguna dapat memasukkan teks *captcha* yang akan ter-*generate* secara otomatis dan berubah-ubah yang akan muncul pada gambar. Pengguna nantinya akan memasukkan hasil *generate* tersebut pada formulir "Silakan Masukkan Captcha". Untuk itu , penerapan *captcha* pada *prototype,* lihat Gambar 8 berikut.

```
<!DOCTYPE html>
<html>
<head>
    <title>Implementasi CAPTCHA</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <h1>Silakan Masukkan CAPTCHA</h1>
    <div id="captcha-container"></div>

    <form>
        <p>CAPTCHA: <input type="text" id="captcha-input"></p>
        <p><input type="button" value="Kirim" onclick="validateCaptcha()"></p>
    </form>

    <script src="captcha.js"></script>
</body>
</html>
```

Gambar 8. Penerapan *captcha* pada *prototype* pengembangan sistem berbasis web

Selanjutnya membuat fungsi yang akan memberikan pola pada penggunaan dan generator tipe *captcha* dengan menggunakan bahasa *javascript* yang banyak dipakai dalam pengembangan web pada dewasa ini. Seperti yang terlihat pada Gambar 9 berikut.

```
function generateCaptcha() {
    const captchaLength = 6;
    const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
    let captcha = '';
    for (let i = 0; i < captchaLength; i++) {
        const randomIndex = Math.floor(Math.random() * characters.length);
        captcha += characters[randomIndex];
    }
    return captcha;
}
function drawCaptcha(captcha) {
    const canvas = document.createElement('canvas');
    const ctx = canvas.getContext('2d');
    canvas.width = 200;
    canvas.height = 60;
    ctx.font = '30px Arial';
    ctx.fillText(captcha, 40, 40);
    document.getElementById('captcha-container').appendChild(canvas);
}
function validateCaptcha() {
    const userInput = document.getElementById('captcha-input').value;
    const captchaValue = sessionStorage.getItem('captcha');
    if (userInput === captchaValue) {
        alert('CAPTCHA Benar, Akses Diberikan.');
    } else {
        alert('CAPTCHA Salah, Akses Ditolak.');
    }
}
document.addEventListener('DOMContentLoaded', function () {
    const captcha = generateCaptcha();
    drawCaptcha(captcha);
    sessionStorage.setItem('captcha', captcha);
});
```

Gambar 9. Pembuatan fungsi *captcha* pada *prototype*

Captcha akan dihasilkan dengan alfanumerik secara akcak pada fungsi *generateCaptcha.* Lalu akan digambar dan dihasilkan menggunakan elemen *canvas* yang terdapat pada fungsi *drawCaptcha.* Kemudian di validasi dan disimpan dalam penyimpanan *session*. Lalu pengguna diarahkan untuk menekan tombol kirim untuk memasukkan *captcha* dan akan dibandingakan dengan *captcha* yang telah di generate. Agar hasil pada *prototype* dapat memiliki tampilan yang jelas, maka dilakukan sedikit penyesuain pada pengaturan *style* pada *prototype* yang ada seperti dibawah ini. Lalu kami menyisipkan jalur untuk penerapan *styling* dengan menggunakan CSS dengan menyisipkan jalur untuk penerapan *styling* dengan menggunakan CSS.

CSS digunakan untuk memberikan tampilan yang lebih menarik bagi pengguna pada halaman website. Semua posisi, jarak, tata letak, huruf, tampilan, dan gaya yang terdapat dalam website akan diatur oleh CSS. Ini akan membantu dalam memberikan pengalaman dan interaksi yang baik terhadap pengguna yang mengkases website tersebut, seperti yang terlihat pada Gambar 10 berikut dimana kami memasukkan unsur *styling* menggunaKn CSS agar tampilan menjadi lebih ramah.

Gambar 10. Penambahan *style* pada *prototype*

PEMBAHASAN

Setelah dilakukan penerapan pada *prototype*, langkah berikutnya adalah melihat hasil yang di *generate* pada tampilan portal mesin pencari. Halaman *web* akan menampilkan teks "Silakan Masukkan *captcha*" di atasnya. Dibawah teks tersebut, akan ditampilkan gambar *captcha* yang dihasilkan secara acak. Gambar tersebut berisi karakter alfanumerik yang harus diidentifikasi oleh pengguna. Dibawah gambar *captcha*, ada kotak input teks yang memungkinkan pengguna untuk memasukkan *captcha* yang mereka lihat pada gambar. Pengguna dapat memasukkan teks *captcha* yang mereka lihat ke dalam kotak input. Ketika tombol "Kirim" ditekan, fungsi akan memeriksa apakah teks yang dimasukkan pengguna sama dengan teks *captcha* yang dihasilkan. Jika sama, maka akan muncul pemberitahuan dengan pesan "*captcha* benar, akses diberikan.". Jika salah, maka akan muncul pemberitahuan dengan pesan "*captcha* salah, akses ditolak.".

Setiap kali halaman dimuat ulang, *captcha* akan terbuat ulang dengan karakter alfanumerik acak yang berbeda, sehingga pengguna akan diberikan *captcha* yang berbeda setiap kali mereka mengakses halaman tersebut. *captcha* dihasilkan secara acak karena untuk menghindari kemungkinan serangan dan percobaan yang mengidentifikasi pola *captcha* yang tetap. Untuk melihat hasil *prototype*, lihat pada Gambar 11.

Gambar 11. Hasil tampilan pada portal halaman pencarian web pada *prototype*

KESIMPULAN

Dalam penelitian ini telah dilakukan dan diterapkan teknik dan metode *captcha*. Hasil dari penelitian ini menunjukkan bahwa penerapan metode *captcha* dengan menggunakan alat pengembangan berbasis web dapat memberikan lapisan keamanan tambahan untuk melindungi situs web dari serangan DDoS dan aktivitas robot atau basis *script* lainnya yang dijalankan otomatis oleh pengguna yang tidak berhak. Dengan membedakan antara pengguna manusia dan pengguna diluar manusia, *captcha* membantu mencegah penyalahgunaan sumber daya situs web dan meningkatkan keamanannya. Namun, untuk keamanan yang lebih optimal, *captcha* harus digunakan dengan berbagai alternatif maupun kombinasi dari penggunaannya, misalnya menggunakan operasi matematika, atau interaksi lain yang misalnya dengan mencocokkan gambar yang identik, atau lain sebagainya. Bisa juga *captcha* digunakan bersamaan dengan metode keamanan lainnya dan dalam strategi perlindungan dan pertahanan dari serangan DDoS. Ada baiknya pengguaan *captcha* juga dibuat dan dilakukan dengan berbagai variasi kombinasi dan rumus pada saat di *generate,* didukung dengan keamanan yang dimplementasikan pada *firewall* dan ataupun jaringan tempat berjalannya sistem berbasis website tersebut. Pada akhirnya untuk keamanan yang optimal, perlu dilakukan pembaharuan secara berkala dalam pengembangan sebuah sistem. Oleh karena itulah perlu adanya perhatian dan evaluasi terhadap setiap penggunaannya secara berkala dan berkelanjutan.

REFERENSI

Andarini, R.Y., Hendradi, P., & Nugroho, S. (2023). Meningkatkan Keamanan Terhadap SQL Injection Studi Kasus Sistem Kepegawaian BNN. *Indonesian Journal of Business Intelegence*. 6(1). 34-42. http://dx.doi.org/10.21927/ijubi.v6i1.3161

Hermawan, R. (2016). Analisis Konsep dan Cara Kerja Serangan Komputer Distributed Denial Of Service. *Faktor Exacta*. 5(1), 1-14. http://dx.doi.org/10.30998/faktorexacta.v5i1.186

Intershop. (2023, June 27). *Concept - Captcha Framework (valid to 7.10)*. Retrieved from intershop.com: https://support.intershop.com/kb/index.php/Display/2B8550

Prasetyo, K.P., Widiyanti, R., & Setyowibowo. S (2015). Keamanan Authentikasi Hotspot Menggunakan Captcha. *Jurnal Teknologi Informasi*. 6(2). 103-114. http://ejurnal.stimata.ac.id/index.php?journal=TI&page=article&op=view&path%5B%5D=165

Kusuma, G. H (2022). Sistem Firewall untuk Pencegahan DDOS ATTACK di Masa Pandemi Covid-19. *Journal of Informatics and Advanced Computing*. 3(1). 52-56. https://journal.univpancasila.ac.id/index.php/jiac/article/view/3853

Juniska, A.A., Padmasari, N.M., Pratiwi, K.W., Listartha, I.M., & Saskara, G.A (2022). Perbandingan DDoS Attack Menggunakan Tools Loic, Hoic, dan Hulk dalam Melakukan Penyerangan pada Suatu Website. *Jurnal Informatika Teknologi dan Sains*. 4(4). 467-471. https://doi.org/10.51401/jinteks.v4i4.2190

Masolo, C. (2023, Feb 28). *Cloudflare Detects a Record 71 Million Request-Per-Second DDoS Attacks*. Retrieved from infoq.com: https://www.infoq.com/news/2023/02/cloudflare-ddos-attack/

Trong, N.D., Huong, T.H., & Hoang, V.T (2023). New Cognitive Deep Learning Captcha. *Journals Sensors MDPI*. 467-471. https://doi.org/10.51401/jinteks.v4i4.2190

Radiyah, U. (2017). Tingkat Kenyamanan Penggunaan Captcha Menggunakan Aplikasi Berbasis Web. *BINA INSANIICT JOURNAL. 4*(2), 169-178. Retrieved from http://ejournal-binainsani.ac.id/index.php/BIICT/article/view/841

Safitri, E.M., Larasati, A.S., & Hari, S.R (2022). Analisis Keamanan Sistem Informasi E-Banking di Era Industri 4.0: Studi Literatur. *Jurnal Ilmiah Teknologi Informasi dan Robotika*. 2(1). 12-16. https://doi.org/10.33005/jifti.v2i1.25

Security, C. (2022). *DDoS Attacks 101: Types, targets, and motivations*. Retrieved from Calyptix Security: https://www.calyptix.com/educational-resources/ddos-attacks-101-types-targets-motivations/

Stewart, D. (2016, November 28). *What Are Denial of Service and DDoS Attacks?* Retrieved from How to Geek: https://www.howtogeek.com/281707/what-are-denial-of-service-and-ddos-attacks/

Sutabri, T. (2012). *Konsep Sistem Informasi.* Yogyakarta: Andi.

Sutabri, T. (2012). *Analisis Sistem Informasi.* Yogyakarta: Andi.

Yoachimik, O. (2022). *DDoS Attack Trends for 2022 Q1*. (Pusiknas Polri) Retrieved July 18, 2023, from https://blog.cloudflare.com/ddos-attack-trends-for-2022-q1/

Pratama, Y., & Sutabri, T (2023). Service Operation ITIL V3 Pada Analisis dan Evaluasi Layanan Teknologi Informasi. *Nuansa Informatika*, 17(1), 169-178. https://doi.org/10.25134/nuansa