SMATIKA : STIKI Informatika Jurnal
Vol. 14, No. 1, Juni 2024 : 111~ 123
ISSN: 2087-0256, e-ISSN: 2580-6939

SQL Injection: Analisis Efektivitas Uji Penetrasi dalam Aplikasi Web

Luthfi Arian Nugraha[1*]
Irwan Alnarus Kautsar[2]
Arif Senja Fitrani[3]
Suprianto[4]

[1,2,3]Informatika, Universitas Muhammadiyah Sidoarjo, Jl. Mojopahit No.666 B, Sidowayah, Celep, Kec. Sidoarjo, Kabupaten Sidoarjo, Jawa Timur, 61215, Indonesia
[1]luthfiarian@umsida.ac.id, [2]irwan@umsida.ac.id, [3]asfjim@umsida.ac.id, [4]suprianto@umsida.ac.id

***Penulis Korespondensi:**
Luthfi Arian Nugraha
luthfiarian@umsida.ac.id

Abstrak

Dalam era digital yang terus berkembang, keamanan sistem informasi menjadi krusial, terutama terhadap serangan SQL Injection yang mengancam integritas data. Penelitian ini bertujuan untuk mengevaluasi kerentanan SQL Injection dalam aplikasi web dan menilai efektivitas metode pengujian penetrasi sebagai alat ukur keamanan. Dengan memanfaatkan tinjauan literatur dan studi terdahulu, penelitian ini mengidentifikasi berbagai teknik serangan dan strategi pertahanan yang digunakan untuk melindungi data. Melalui pengujian penetrasi yang sistematis pada sepuluh situs web, penelitian ini menghasilkan data performa yang mencerminkan tingkat keberhasilan serangan dan waktu yang dibutuhkan untuk penetrasi. Hasil menunjukkan variasi dalam efektivitas tools pengujian penetrasi, dengan beberapa situs menunjukkan kerentanan yang signifikan. Untuk meningkatkan keamanan aplikasi web, penelitian ini menyarankan pembaruan bahasa pemrograman, penerapan paradigma OOP dan MVC, penggunaan Rest API, implementasi WAF, dan penggunaan CAPTCHA. Temuan ini memberikan wawasan untuk pengembangan strategi keamanan yang lebih tangguh dan adaptif dalam menghadapi ancaman siber.
Kata Kunci: Keamanan Siber; Injeksi SQL; Uji Penetrasi; Kerentanan Web

Abstract

In the continuously evolving digital era, information system security becomes crucial, particularly against SQL Injection attacks that threaten data integrity. This research aims to evaluate the vulnerability to SQL Injection in web applications and assess the effectiveness of penetration testing methods as a security measure. Utilizing a literature review and previous studies, this research identifies various attack techniques and defense strategies used to protect data. Through systematic penetration testing on ten websites, this study produces performance data reflecting the success rate of attacks and the time required for penetration. The results show variations in the effectiveness of penetration testing tools, with some sites exhibiting significant vulnerabilities. To enhance the security of web applications, this research suggests updating programming languages, implementing OOP and MVC paradigms, using Rest APIs, implementing WAFs, and utilizing CAPTCHAs. These findings provide insights for developing more robust and adaptive security strategies in the face of cyber threats.
Keywords: Cybersecurity; SQL Injection; Penetration Testing; Web Vulnerability

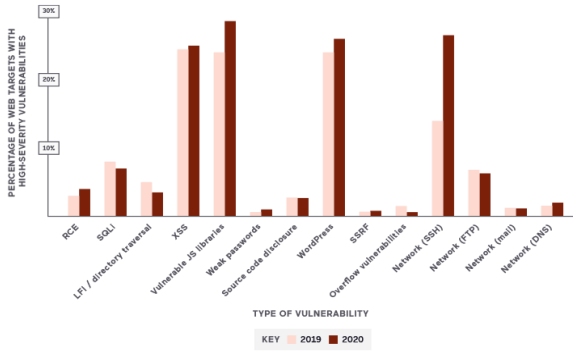
Cite: Nugraha, L.A., dkk. (2024). SQL Injection: Analisis Efektivitas Uji Penetrasi dalam Aplikasi Web. SMATIKA: STIKI Informatika Jurnal,14(1). doi: https://doi.org/10.32664/smatika.v14i01.1224

1. Pendahuluan

Teknologi sistem informasi saat ini berkembang sangat pesat. Informasi yang mudah diakses di internet secara gratis dan cepat tanpa batasan membuat banyak data terbuka yang mudah diakses oleh semua orang. Hal ini menyebabkan lemahnya integritas data asli yang dipresentasikan, menjadi tidak otentik atau telah diubah oleh pengguna yang tidak bertanggung jawab[1].

Sistem informasi menjadi suatu kebutuhan sehari-hari masyarakat, salah satu contohnya yaitu situs web berita, media sosial, dan blog yang bersifat individu. Informasi yang disuguhkan ialah hasil olah data dari setiap pemilik atau instansi yang menyediakan platform tersebut[2].

SQL Injection merupakan teknik serangan eksploitasi pada basis data yang terintegrasi dengan situs web, yang memungkinkan penghancuran dan manipulasi data[3]. Serangan eksploitasi tersebut menyebabkan kerusakan bahkan kehilangan data yang disimpan pada situs web. Implementasi teknik serangan ini dapat dilakukan melalui *Uniform Resource Locator* (URL) pada situs web menggunakan *tools* yang bersifat sumber terbuka[4].



Gambar 1. *Statistik Acunetix pada Musim Semi Tahun 2021*

Laporan tahunan Acunetix yang terbit di awal tahun 2021 menunjukkan bahwa serangan *SQL Injection* masih menjadi kerentanan yang sering ditemui, dengan angka kejadian sekitar 8% di tahun 2019 dan menurun sedikit menjadi 7% di tahun 2020. Informasi ini tergambar dalam Gambar 1 yang menampilkan daftar kerentanan tersebut. Kendati mengalami penurunan sebesar 1% antara tahun 2019 dan 2020, serangan *SQL Injection* tetap berada di urutan kelima dalam daftar 14 jenis serangan yang paling umum[5].

Penetrasi merupakan salah satu metode uji pengukuran terjadinya kerentanan sebuah aplikasi yang telah dibangun atau dalam tahapan pembuatan[6]. Uji penetrasi dapat menjaga integritas sebuah data setiap pengguna dari aplikasi yang digunakan[7].

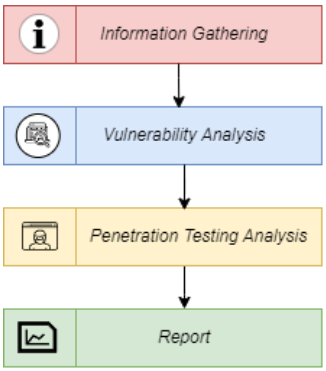
Penelitian pertama [8] dengan judul "*Web Application Penetration Testing Using SQL Injection Attack*" melakukan penelitian dengan menggunakan metode injeksi SQL digunakan untuk mengidentifikasi kerentanan injeksi SQL dalam aplikasi web. Metode ini mengeksploitasi kerentanan keamanan dalam aplikasi web untuk mengirimkan perintah SQL yang tidak valid. Jika kerentanan ini ada maka perintah SQL yang tidak valid akan dijalankan oleh database, yang dapat menyebabkan serangan injeksi SQL. Tujuan umum pada penelitian tersebut adalah mengidentifikasi dan mengeksploitasi kerentanan *SQL injection* pada aplikasi web.

Penelitian kedua [9] dengan judul "*SQL Injection Attacks Countermeasures Assessment*" dengan menggunakan metode identifikasi countermeasure pada serangan eksploitasi injeksi SQL. Tujuan umum pada penelitian tersebut adalah mengevaluasi efektivitas countermeasure terhadap serangan *SQL injection*.

Penelitian Keempat [10] dengan judul *"Query Response Time Comparison SQL and No SQL for Contact Tracing Application"* menggunakan metode eksperimen untuk melakukan komparasi terhadap dua jenis basis data relasional (SQL) dan non-relasional (NoSQL). Tujuan penelitian yang terkait yaitu menginformasikan pada pengembang tentang perbandingan basis data relasional memiliki struktur data yang lebih kompleks dan efisien untuk merepresentasikan hubungan antar data, sedangkan basis data non-relasional memiliki struktur data yang lebih sederhana dan efisien untuk merepresentasikan data dalam jumlah besar.

2. Metode Penelitian

Penelitian ini menggunakan metode pengujian penetrasi. Gambar 2 menunjukkan tahapan dalam melakukan penelitian[11].



Gambar 2. *Tahapan Metode Uji Penetrasi*

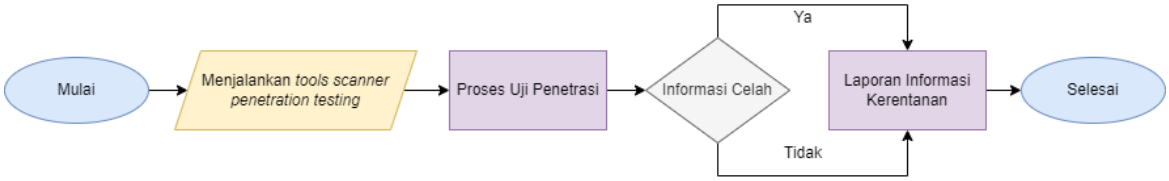
Information gathering adalah fase awal untuk mendapatkan sebuah informasi pada platform yang dituju. Tujuan dari pengumpulan informasi adalah untuk mendapatkan data sensitif dari target yang akan diteliti[12].

Vulnerability analysis tahapan penting analisis dalam melakukan pengujian penetrasi. Tahapan ini menentukan metode pengujian yang akan digunakan. Dengan demikian, akan diperoleh data mentah yang dapat diproses menggunakan *tools* yang telah disediakan[13].

Penetration testing analysis merupakan tahapan implementasi pengujian penetrasi pada platform yang dituju Tahapan ini akan menghasilkan data kinerja sebagai hasil dari laporan pengujian penetrasi[14].

Report atau penyusunan laporan adalah tahapan terakhir yang melibatkan penyajian data yang telah diproses sehingga dapat dipahami oleh pembaca. Tahapan ini juga melibatkan penyajian informasi yang berasal dari data yang telah diperoleh[15].

Diagram alir atau *flowchart* merupakan representasi alur prosedur penelitian ini yang dilakukan secara sistematis[16]. Dengan menggunakan diagram alir tersebut, rangkaian proses uji penetrasi dapat dilakukan lebih mudah.



Gambar 3. *Diagram Alir Pengujian Penetrasi*

Gambar 3 adalah diagram alir yang menunjukkan alur pengujian penetrasi, diikuti dengan pelaksanaan pengujian penetrasi pada platform yang dituju. Selanjutnya, informasi tentang celah keamanan diperoleh dan dicatat dalam laporan yang mencakup status keberhasilan dan waktu yang diperlukan selama pengujian penetrasi[17].

3. Hasil

Peneliti akan melakukan simulasi serangan pada 10 situs web, terdiri dari 5 situs web sumber terbuka yang sudah memiliki izin dan 5 situs web secara acak seperti pada tabel 1 tersebut untuk menemukan kelemahan atau celah keamanan yang dapat dimanfaatkan oleh penyerang untuk melakukan tindakan berbahaya, seperti mencuri data sensitif, merusak sistem, atau mengganggu operasi situs web[15].

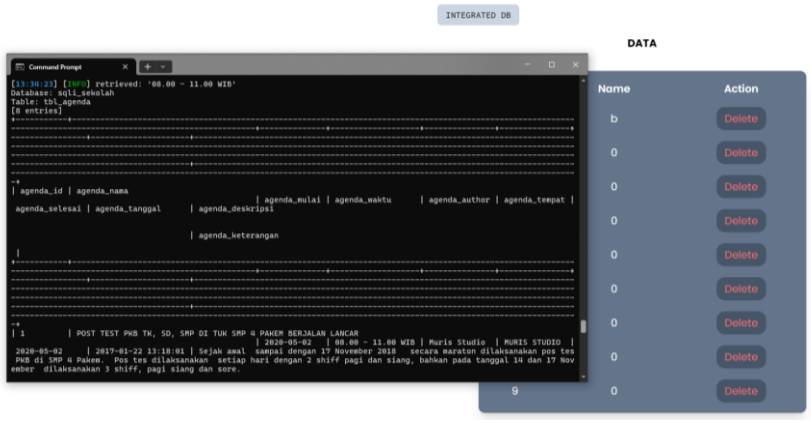
Tabel 1. *Daftar Situs Web dan Analisis Kerentanannya*

Nama Situs Web	Jenis Situs Web	Sasaran	Metode
Situs-A	Pendidikan Negeri	Form	POST
Situs-B	Pendidikan Swasta	Form	POST
Situs-C	Hiburan	Login Admin	POST
Situs-D	Pekerja Lepas A	Form	POST
Situs-E	Pekerja Lepas B	Form	POST
Situs-F	Bisnis	Pencarian	GET
Situs-G	Penyedia Layanan	Pencarian	GET
Situs-H	Blog Komersial	Pencarian	POST
Situs-I	Blog Individu	Form	GET
Situs-J	Forum Komunitas	Form	GET

Situs web yang telah dikumpulkan, kemudian dilakukan analisis terhadap sasaran yang digunakan dan metode yang diterapkan, sebagaimana dirangkum dalam tabel 1[18].

Penelitian ini mengimplementasikan tiga alat uji penetrasi yang bersifat sumber terbuka. Pemilihan alat-alat ini didasarkan pada beberapa pertimbangan krusial. Pertama, sifat sumber terbuka dari alat-alat ini memungkinkan kemudahan dalam implementasi, yang vital dalam konteks penelitian keamanan siber. Kedua, dokumentasi yang lengkap untuk setiap alat memastikan bahwa proses uji penetrasi dapat dilakukan dengan standar yang konsisten dan dapat diulang. Terakhir, dukungan komunitas yang kuat untuk alat-alat ini memberikan jaminan tambahan dalam hal pembaruan keamanan dan resolusi masalah yang cepat. Berikut analisis dari setiap *Tools* dalam pengujian penetrasi pada 10 situs web pada tabel 1.

Data penetrasi merupakan komponen penting dalam pengujian, khususnya pada pengujian penetrasi yang bertujuan untuk mengidentifikasi kerentanan dan celah keamanan dalam sistem informasi. Data ini diperoleh melalui penggunaan berbagai *tools*, salah satunya adalah *Tools* A yang tercantum dalam tabel 3.



Gambar 4. *Contoh Uji Penetrasi Menggunakan Tools A*

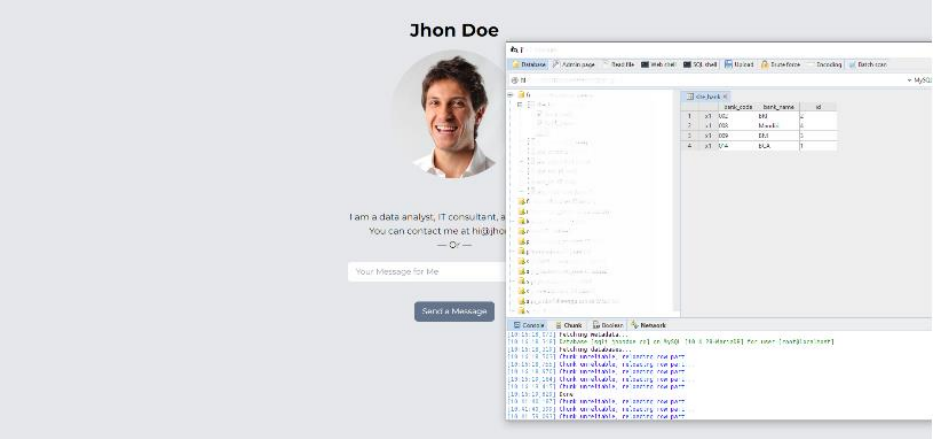
Tabel 2*. Hasil Uji Penetrasi Tools A*

Nama Situs Web	Waktu	Status Keberhasilan
Situs-A	2 menit 23 detik	Ya
Situs-B	1 menit 36 detik	Ya
Situs-C	8 menit 27 detik	Tidak
Situs-D	28 menit 22 detik	Tidak
Situs-E	7 menit 13 detik	Tidak
Situs-F	1 menit 27 detik	Tidak
Situs-G	7 menit 24 detik	Tidak
Situs-H	2 menit 44 detik	Tidak
Situs-I	49 Detik	Ya
Situs-J	10 menit 45 detik	Ya

Pengujian selanjutnya, penetrasi dilakukan menggunakan *Tools* B. Data penetrasi yang dihasilkan kemudian dianalisis untuk mengevaluasi ketahanan sistem terhadap serangan siber.

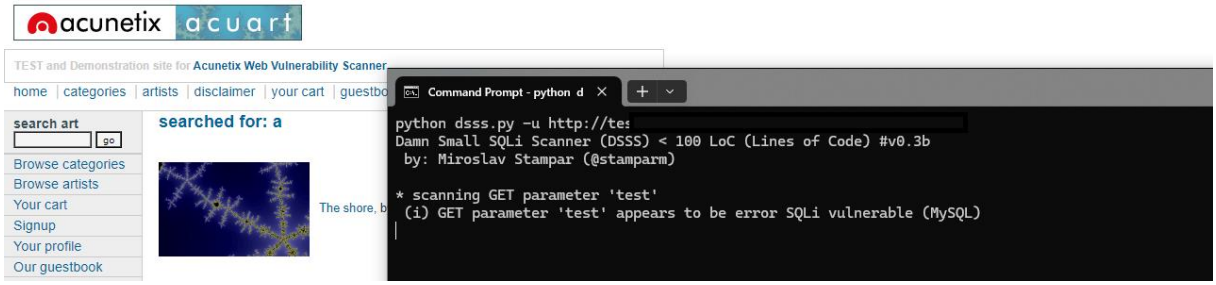
Tabel 3. *Hasil Uji Penetrasi Tools B*

Nama Situs Web	Waktu	Status Keberhasilan
Situs-A	8 detik	Tidak
Situs-B	4 menit 18 detik	Ya
Situs-C	1 detik	Tidak
Situs-D	4 menit 12 detik	Tidak
Situs-E	2 detik	Tidak
Situs-F	4 menit 51 detik	Tidak
Situs-G	6 menit 8 detik	Tidak
Situs-H	1 detik	Tidak
Situs-I	1 menit 22 detik	Ya
Situs-J	52 detik	Ya



Gambar 5. *Contoh Implementasi Tools B*

Pengujian terakhir dilakukan dengan memanfaatkan *Tools* C, sebuah perangkat lunak yang dikembangkan untuk memindai situs web dengan cara menulis dalam kurang dari 100 baris kode berbahaya.



Gambar 6. *Contoh Uji Penetrasi Menggunakan Tools C*

Tabel 4. *Hasil Uji Penetrasi Tools C*

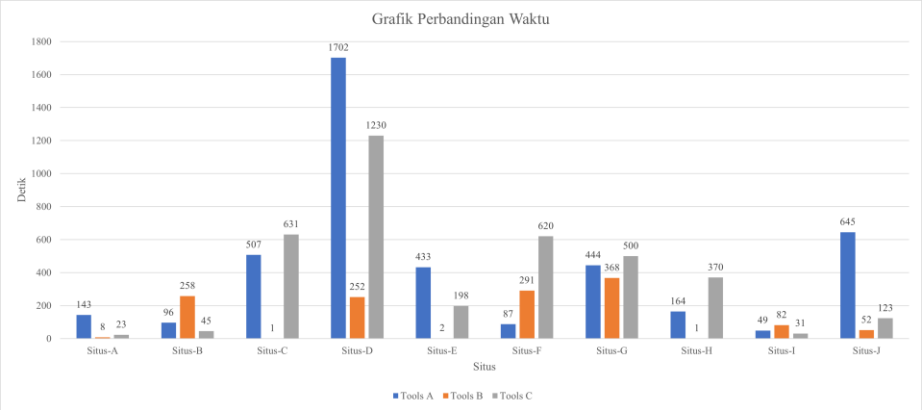
Nama Situs Web	Waktu	Status Keberhasilan
Situs-A	23 detik	Ya
Situs-B	45 detik	Ya
Situs-C	10 menit 31 detik	Tidak
Situs-D	20 menit 30 detik	Tidak
Situs-E	3 menit 18 detik	Tidak
Situs-F	10 menit 20 detik	Tidak
Situs-G	8 menit 20 detik	Tidak
Situs-H	6 menit 10 detik	Tidak
Situs-I	31 detik	Ya
Situs-J	2 menit 3 detik	Ya

4. Pembahasan

Data uji penetrasi yang presentasikan pengujian menggunakan *Tools* A menunjukan bahwa 4 dari 10 situs web berhasil melemahkan (Situs-A, Situs-B, Situs-I, dan Situs-J). *Tools* B menunjukkan bahwa 3 dari 10 situs web berhasil ditembus (Situs-B, Situs-I, dan Situs-J). *Tools* C menunjukkan bahwa 4 dari 10 situs web berhasil ditembus (Situs-A, Situs-B, Situs-I, dan Situs-J).

Situs-C hingga Situs-H menunjukkan ketahanan yang baik terhadap serangan siber, karena berhasil lolos dalam semua pengujian penetrasi, ketahanan ini dapat diatributkan kepada implementasi proteksi berlapis yang terintegrasi selama fase pengembangan situs web. Situs-B, Situs-I, dan Situs-J menunjukkan kerentanan yang tinggi terhadap serangan siber, yang sebagian besar disebabkan oleh penggunaan teknik pengembangan prosedural sehingga memudahkan uji penetrasi.

Fokus utama dari penelitian ini adalah untuk mengidentifikasi dan mengeksplorasi titik celah yang memungkinkan serangan siber untuk menyusup dan mengirimkan kode berbahaya. Tujuan akhir adalah untuk mendapatkan akses ke basis data situs web yang ditargetkan melalui *Tools* yang digunakan dalam penelitian ini. Sebagai bagian dari analisis penelitian, gambar 7 menyajikan grafik perbandingan waktu yang dibutuhkan untuk menembus masing-masing situs web, memberikan visualisasi yang efektif dari data yang dikumpulkan.



Gambar 7. *Grafik Hasil Uji Penetrasi Berdasarkan Waktu*

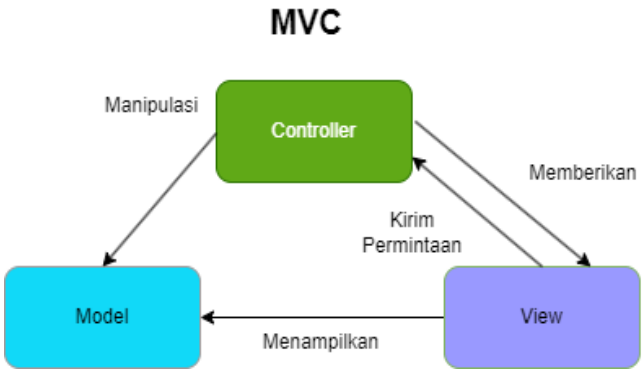
Solusi pertama dari hasil uji penetrasi adalah menggunakan bahasa pemrograman dengan versi terbaru. Dalam konteks pengembangan perangkat lunak, pembaruan bahasa pemrograman merupakan salah satu strategi krusial untuk mengamankan aplikasi dari serangan injeksi SQL. Pembaruan ini tidak hanya memperbaiki kerentanan yang telah dikenali, tetapi juga memperkenalkan fitur keamanan yang lebih canggih. Versi terbaru bahasa pemrograman, seperti PHP 8.3 yang dirilis pada 11 April 2024, kerap dilengkapi dengan mekanisme pertahanan yang ditingkatkan, termasuk sanitasi *input* yang lebih efektif dan dukungan yang lebih baik untuk *prepared statements*. Fitur-fitur ini secara signifikan mengurangi risiko serangan injeksi SQL dengan membatasi kemungkinan eksekusi perintah SQL yang berbahaya[19]. Selain itu, pembaruan bahasa pemrograman memastikan kepatuhan terhadap standar keamanan terkini dan mendukung inisiatif pemantauan keamanan yang berkelanjutan. Oleh karena itu, pembaruan ke versi terbaru tidak hanya merupakan praktik keamanan yang baik, tetapi juga bagian integral dari manajemen risiko yang proaktif dalam pengembangan aplikasi web .

Solusi kedua adalah penerapan metode *object oriented programming* (OOP). Pemrograman yang berorientasi pada objek adalah metode pengembangan perangkat lunak yang memusatkan fokus pada objek sebagai elemen kunci dalam proses pembuatan program. Dalam OOP, objek didefinisikan oleh atributnya, yang berupa data, dan metodenya adalah fungsi yang menentukan perilaku objek tersebut. Peningkatan keamanan aplikasi dengan enkapsulasi data, pemisahan kepentingan, dan pola desain seperti DAO dan *Repository* yang memfasilitasi penggunaan *prepared statements* dan *parameterized queries*. Pendekatan ini memungkinkan kontrol akses yang ketat dan validasi input yang efektif, mengurangi risiko serangan *SQL Injection* secara signifikan, seperti contoh kode PHP OOP sederhana pada Gambar 9[20][21].

```php
<?php

class Hewan {
  public $nama;

  function __construct($nama) {
    $this->nama = $nama;
  }

  function suara() {
    echo "Suara hewan ini tidak diketahui";
  }
}

class Kucing extends Hewan {
  function suara() {
    echo "Meaong!";
  }
}

class Anjing extends Hewan {
  function suara() {
    echo "Guk! Guk!";
  }
}

$kucing = new Kucing("Garfield");
$anjing = new Anjing("Siberian Husky");

echo "Hewan bernama " . $kucing->nama . " bersuara: "; $kucing->suara();

echo "\nHewan bernama " . $anjing->nama . " bersuara: "; $anjing->suara();
```

Gambar 8. *Contoh Kode Bahasa Pemrograman PHP OOP Sederhana*

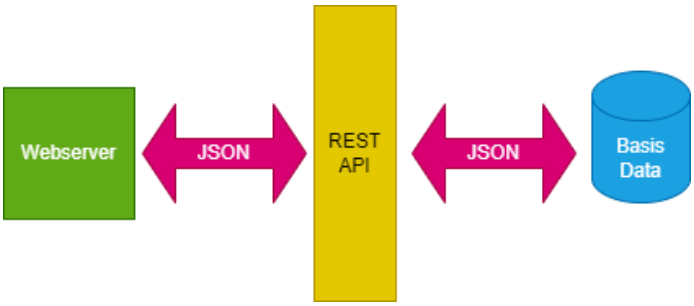
Solusi ketiga adalah penerapan metode *Model-View-Controller* (MVC). MVC adalah model arsitektur perangkat lunak yang efektif dan dapat diadaptasi, yang tidak hanya berguna untuk mengembangkan berbagai jenis aplikasi tetapi juga menawarkan keuntungan seperti pemisahan fungsi yang jelas, keterkaitan yang minimal antar komponen, kemudahan dalam pengujian, dan sederhana untuk digunakan. MVC mendukung keamanan aplikasi dengan memungkinkan pemisahan yang jelas antara logika bisnis dan antarmuka pengguna, yang dapat membantu meminimalisir risiko serangan, sehingga memperkuat aplikasi terhadap ancaman keamanan[22]. Gambar 10 menunjukkan berkas tampilan, kontrol, dan komunikasi dengan basis data secara terpisah. Hal ini mencegah *tools* mendeteksi instruksi kode pada komunikasi terhadap basis data yang dituju.



Gambar 9. *Diagram MVC*

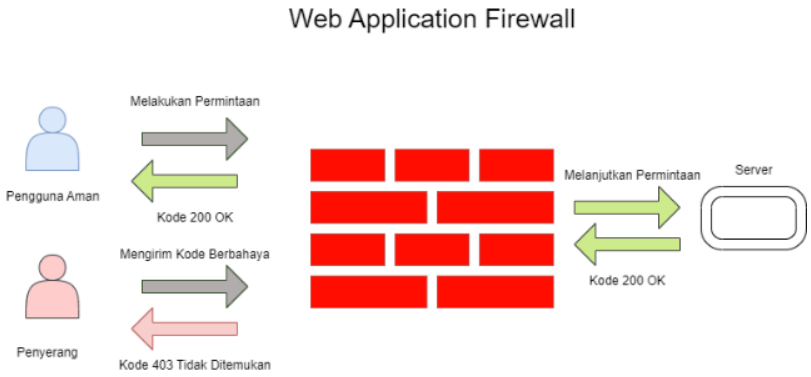
Solusi keempat adalah penerapan Rest API. *Representational State Transfer Application Programming Interface* (Rest API) adalah sebuah antarmuka pemrograman aplikasi yang digunakan untuk memfasilitasi komunikasi dan pertukaran data antar berbagai perangkat lunak atau sistem dalam jaringan komputer seperti diagram pada gambar 11. Rest API berfungsi sebagai perantara yang mengikuti seperangkat aturan dan batasan untuk memastikan komunikasi yang handal dan mudah digunakan. Rest API merupakan gaya arsitektur yang umum digunakan dalam pengembangan perangkat lunak untuk memungkinkan aplikasi berinteraksi satu sama lain melalui protokol *Hypertext Transfer Protocol* (HTTP), yang merupakan protokol dasar yang digunakan untuk komunikasi web. Rest API menggunakan metode HTTP standar

seperti *get, post, put*, dan *delete* untuk melakukan operasi *create, read, update, delete* (CRUD) pada data[23]. Rest API juga dapat menggunakan format data umum seperti *Javascript Object Notation* (JSON) atau *Extensible Markup Language* (XML) untuk bertukar data dengan mudah antara aplikasi yang berbeda. Hal ini menjadikan rest api sebagai pilihan yang populer untuk membangun layanan web yang fleksibel dan dapat diskalakan dan tidak dapat mengeksekusi intruksi bahasa pemrogaman basis data seperti *tools* yang digunakan pada uji penetrasi[24].



Gambar 10. *Diagram Komunikasi Rest API*

Solusi kelima adalah implementasi perangkat lunak *Web Application Firewall* (WAF). WAF merupakan sistem pengamanan yang secara khusus dikembangkan untuk mengamankan aplikasi web dari ancaman serangan siber seperti *SQL Injection*, *Cross Site Scripting* (XSS), dan *Distributed Denial-of-Service* (DDoS). Berfungsi sebagai penyaring, WAF mengawasi serta mengevaluasi lalu lintas data yang menuju aplikasi web, mengidentifikasi kegiatan yang mencurigakan, serta menghalanginya agar tidak sampai ke tujuan[25]. Seperti yang dicontohkan pada gambar 11, *tools* yang digunakan dalam uji penetrasi terdeteksi sebagai penyerangan kedalam sistem situs web.



Gambar 11. *Diagram Kerja WAF*

Solusi keenam adalah implementasi metode Captcha. Serangan *SQL Injection*, XSS, dan DDoS dapat dicegah menggunakan metode *Completely Automated Public Turing test to tell Computers and Humans Apart* (CAPTCHA). Captcha adalah sebuah fitur keamanan pada situs web yang berfungsi sebagai pendeteksi pengguna manusia dan komputer, jika sistem mendeteksi adanya kecurangan seperti menggunakan *tools* berbahaya, maka yang terjadi yaitu memblokir Internet Protocol (IP) pada pengguna yang melakukan kecurangan[26].

Pengujian yang dilakukan, *Tools* A, B, dan C menunjukkan variasi dalam efektivitas mereka. Tools A tampaknya lebih efektif dalam menembus sejumlah situs web, sementara *Tools* B dan C memiliki tingkat keberhasilan yang lebih rendah. Hal ini dapat menunjukkan bahwa *Tools* A memiliki kemampuan yang lebih baik dalam mengidentifikasi dan mengeksploitasi kerentanan yang ada.

Situs-A dan Situs-B menunjukkan konsistensi dalam keberhasilan penetrasi menggunakan *Tools* A dan B, namun *Tools* C gagal menembus Situs-A. Ini menimbulkan pertanyaan tentang keandalan dan metodologi yang digunakan oleh *Tools* C dalam pengujian.

Situs-I dan Situs-J berhasil ditembus oleh semua *tools*, yang menandakan bahwa kedua situs ini memiliki kerentanan yang lebih serius. Ini memerlukan tindakan segera untuk menambal celah keamanan yang ada.

Waktu yang dibutuhkan untuk menembus setiap situs web sangat bervariasi, yang menunjukkan perbedaan dalam tingkat keamanan dan kompleksitas setiap situs. Situs dengan waktu penetrasi yang lebih singkat mungkin memiliki kerentanan yang lebih mudah diidentifikasi atau kurangnya langkah-langkah keamanan yang efektif.

Hasil pengujian ini juga memberikan wawasan penting untuk pengembangan *tools* penetrasi di masa depan. *Tools* yang dapat menyesuaikan pendekatannya berdasarkan karakteristik unik dari setiap situs web mungkin lebih efektif dalam mengidentifikasi kerentanan.

Teknologi yang digunakan untuk pengembangan di setiap situs web memiliki pengaruh besar dalam penelitian ini. Sebagaimana contohnya dalam Situs-A dan Situs-I menggunakan teknologi pengembangan prosedural yang memiliki kerentanan sangat tinggi, dibandingkan dengan Situs-D dan Situs-E yang menerapkan teknologi pengembangan MVC sebagai pemrosesan data yang akan ditampilkan kepada pengguna situs dan Rest API sebagai cara kerja komunikasi aplikasi web dengan basis data.

Penelitian ini melibatkan uji penetrasi yang dilakukan pada 10 situs web terpilih. Langkah ini diambil untuk menguji efektivitas metode penetrasi dalam mengidentifikasi kerentanan sistem. Meskipun waktu yang tersedia untuk uji penetrasi ini terbatas, prosedur yang digunakan telah dirancang untuk memastikan bahwa hasil yang diperoleh dapat memberikan wawasan yang signifikan mengenai keamanan situs web yang diuji.

Penelitian ini membatasi penggunaan *tools* penetrasi ke tiga *tools* utama. Pembatasan ini dilakukan dengan pertimbangan khusus untuk memastikan konsistensi dan keandalan data. Ketiga *tools* tersebut dipilih berdasarkan kemampuan *tools* untuk memberikan wawasan komprehensif tentang keefetivitasan terhadap serangan *SQL Injection*. Selain itu, penggunaan alat yang lebih terbatas memungkinkan peneliti untuk melakukan analisis yang lebih mendalam dan spesifik, yang pada gilirannya meningkatkan kualitas hasil penelitian. Meskipun penggunaan lebih banyak alat dapat memberikan variasi data yang lebih luas, namun hal itu juga dapat menyebabkan kompleksitas dan variabilitas yang tidak diinginkan, yang dapat mengaburkan temuan utama penelitian.

Peneliti menghadapi beberapa tantangan, termasuk keterbatasan informasi mengenai teknologi yang digunakan selama pengembangan situs web yang diuji secara acak. Akibatnya, pendekatan yang diterapkan adalah *Blackbox Testing*, di mana berfokus pada analisis input dan output dari berbagai alat tanpa akses ke struktur kode internal situs. Metode ini memungkinkan kami untuk menilai keamanan eksternal situs web tanpa memerlukan pengetahuan mendalam tentang arsitektur sistemnya.

Penelitian ini menggunakan 3 *tools* dan setiap penggunaan *tools* tersebut memiliki tingkatnya masing-masing. *Tools* A memiliki tingkat menengah dalam melakukan uji penetrasi, tidak ada tampilan yang ramah pengguna seperti *Graphic User Interface* (GUI). *Tools* B memiliki tingkat menengah dan lanjut, meskipun *tools* ini terdapat GUI untuk melakukan uji penetrasi, tetapi butuh pengaturan khusus untuk menjalankan *tools* tersebut. *Tools* C memiliki pengalaman yang ramah pengguna, tidak memiliki GUI bukan menjadi masalah pada *tools* tersebut, dengan satu kali instruksi *tools* tersebut melakukan tugasnya dengan mudah.

5. Penutup

Serangan *SQL Injection* masih menjadi ancaman serius bagi situs web. Penelitian ini menunjukkan bahwa pengujian penetrasi merupakan metode yang efektif untuk mengidentifikasi kerentanan dan celah keamanan dalam sistem informasi. Namun, efektivitas *tools* penetrasi bervariasi, dan beberapa situs web memiliki kerentanan yang lebih serius dibandingkan situs web lainnya.

Penelitian ini memiliki implikasi langsung terhadap industri teknologi informasi, khususnya dalam pengembangan dan pemeliharaan situs web. Dengan mengidentifikasi efektivitas berbagai *tools* penetrasi, penelitian ini memberikan wawasan penting bagi pengembang perangkat lunak dalam memilih *tools* yang paling sesuai untuk melindungi aset digital mereka. Lebih lanjut, temuan ini mendorong industri untuk mengadopsi praktik terbaik dalam pengembangan perangkat lunak, seperti penggunaan bahasa pemrograman terbaru dan arsitektur yang lebih aman seperti OOP, MVC, dan Rest API.

Keamanan situs web memiliki dampak yang luas terhadap masyarakat, mengingat ketergantungan kita yang semakin meningkat pada layanan daring untuk berbagai aspek kehidupan sehari-hari. Dengan mengurangi risiko serangan *SQL Injection*, penelitian ini berkontribusi pada perlindungan data pribadi pengguna, yang pada gilirannya meningkatkan kepercayaan masyarakat terhadap teknologi digital. Ini juga membantu dalam membangun lingkungan daring yang lebih aman, di mana transaksi keuangan dan pertukaran informasi dapat dilakukan tanpa kekhawatiran akan pencurian identitas atau kehilangan data.

Hasil penelitian ini menekankan pentingnya penerapan langkah-langkah keamanan yang komprehensif, termasuk WAF dan Captcha, untuk melindungi situs web dari serangan yang semakin canggih. Ini tidak hanya membantu pengembang situs web dalam meningkatkan keamanan produk mereka tetapi juga memberikan arah bagi peneliti keamanan untuk mengembangkan solusi pencegahan yang lebih inovatif dan efektif terhadap serangan *SQL Injection*.

Dalam penelitian ini, metode pengujian penetrasi yang sistematis telah berhasil mengungkap kerentanan terhadap serangan *SQL Injection* pada aplikasi web, menyoroti pentingnya pembaruan keamanan dan penerapan paradigma pemrograman modern. Sebagai saran untuk penelitian selanjutnya, akan bermanfaat untuk mengeksplorasi integrasi teknologi kecerdasan buatan dalam pengujian penetrasi untuk respons yang lebih dinamis terhadap ancaman siber. Meskipun penelitian ini memberikan wawasan berharga, terdapat keterbatasan dalam variasi *tools* pengujian dan sampel situs web yang terbatas, yang menyarankan perlunya pendekatan yang lebih inklusif dan diversifikasi alat pengujian untuk memperoleh perspektif yang lebih luas dalam keamanan siber. Keberhasilan dalam identifikasi kerentanan ini menegaskan kontribusi signifikan penelitian terhadap pengembangan strategi keamanan yang lebih tangguh, sementara keterbatasannya membuka peluang untuk peningkatan dan inovasi berkelanjutan dalam bidang keamanan informasi.

Referensi

[1] H. Wakkang and B. Irianto, "IMPLEMENTASI WEB SERVICE DENGAN METODE REST API UNTUK INTEGRASI DATA COVID 19 DI SULAWESI SELATAN," *Jurnal Sintaks Logika (JSilog) Jurnal Penelitian Ilmiah Teknik Informatika*, vol. 2, no. 1, pp. 12–22, 2022, doi: 10.31850/jsilog.v2i1.

[2] M. A. Z. Risky and Y. Yuhandri, "Optimalisasi dalam Penetrasi Testing Keamanan Website Menggunakan Teknik SQL Injection dan XSS," *Jurnal Sistim Informasi dan Teknologi*, pp. 215–220, Aug. 2021, doi: 10.37034/jsisfotek.v3i4.68.

[3] Abdul Djalil Djayali, "Analisa Serangan SQL Injection pada Server pengisian Kartu Rencana Studi (KRS) Online," *JAMINFOKOM*, vol. 1, 2020.

[4] P. Gio *et al.*, "Analisis Perbandingan Tools SQL Injection Menggunakan SQLmap, SQLsus dan The Mole," *Informatik : Jurnal Ilmu Komputer*, vol. 18, p. 2022, 2022.

[5] Invicti, "The Invicti AppSec Indicator Spring 2021 Edition: Acunetix Web Vulnerability Report," Acunetix. Accessed: Dec. 29, 2023. [Online]. Available: https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2021/

[6] A. Faidlatul Habibah, F. Shabira, and I. Irwansyah, "Pengaplikasian Teori Penetrasi Sosial pada Aplikasi Online Dating," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 3, no. 1, pp. 44–53, Jan. 2021, doi: 10.47233/jteksis.v3i1.183.

[7] S. U. Sunaringtyas, D. Surya Prayoga, J. K. Siber, P. Siber, and S. Negara, "Edu Komputika Journal Implementasi Penetration Testing Execution Standard Untuk Uji Penetrasi Pada Layanan Single Sign-On," 2021. [Online]. Available: http://journal.unnes.ac.id/sju/index.php/edukom

[8] A. Alanda, D. Satria, M. Isthofa Ardhana, A. A. Dahlan, and A. Mooduto, "Web Application Penetration Testing Using SQL Injection Attack," *JOIV : International Journal on Informatics Visualization*, vol. 5, no. 3, 2021, [Online]. Available: www.joiv.org/index.php/joiv

[9] M. Alenezi, M. Nadeem, and R. Asif, "SQL injection attacks countermeasures assessments," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, pp. 1121–1131, Feb. 2020, doi: 10.11591/ijeecs.v21.i2.pp1121-1131.

[10] A. B. Setyawan, I. A. Kautsar, and N. L. Azizah, "Query Response Time Comparison SQL and No SQL for Contact Tracing Application," *PELS*, vol. 2, no. 2, 2022.

[11] M. Hasibuan and A. M. Elhanafi, "Penetration Testing Sistem Jaringan Komputer Menggunakan Kali Linux untuk Mengetahui Kerentanan Keamanan Server dengan Metode Black Box," *sudo Jurnal Teknik Informatika*, vol. 1, no. 4, pp. 171–177, Dec. 2022, doi: 10.56211/sudo.v1i4.160.

[12] C. Budi Setiawan, D. Hariyadi, A. Sholeh, A. Wisnuaji, A. Yani Yogyakarta, and P. Widya Adijaya Nusantara, "Pengembangan Aplikasi Information Gathering Berbasis HybridApps," *Jurnal Informatika dan Teknologi Informasi (INTEK)*, vol. 5, 2022.

[13] A. Zirwan, "Pengujian dan Analisis Kemanan Website Menggunakan Acunetix Vulnerability Scanner," *Jurnal Informasi dan Teknologi*, pp. 70–75, Mar. 2022, doi: 10.37034/jidt.v4i1.190.

[14] Y. A. Pohan, "Meningkatkan Keamanan Webserver Aplikasi Pelaporan Pajak Daerah Menggunakan Metode Penetration Testing Execution Standar," *Jurnal Sistim Informasi dan Teknologi*, pp. 1–6, Mar. 2021, doi: 10.37034/jsisfotek.v3i1.36.

[15] J. Panjaitan and A. F. Pakpahan, "Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, Apr. 2021, doi: 10.28932/jutisi.v7i1.3098.

[16] Syamsiah Syamsiah, "Perancangan Flowchart dan Pseudocode Pembelajaran Mengenal Angka dengan Animasi untuk Anak PAUD Rambutan," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 4, no. 1, 2019.

[17] S. T. Argaw *et al.*, "Cybersecurity of Hospitals: Discussing the challenges and working towards mitigating the risks," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1. BioMed Central Ltd, Jul. 03, 2020. doi: 10.1186/s12911-020-01161-7.

[18] Fuad Dwi Hanggara and R. D. E. Putra, "Analisis Sistem Antrian Pelanggan SPBU Dengan Pendekatan Simulasi Arena," *Jurnal INTECH Teknik Industri Universitas Serang Raya*, vol. 6, no. 2, pp. 155–162, Dec. 2020, doi: 10.30656/intech.v6i2.2543.

[19] "PHP: PHP 8.3.0 Release Announcement," PHP. Accessed: Apr. 21, 2024. [Online]. Available: https://www.php.net/releases/8.3/en.php

[20] D. P. Y. Ardiana and L. H. Loekito, "Gamification design to improve student motivation on learning object-oriented programming," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2020. doi: 10.1088/1742-6596/1516/1/012041.

[21] M. Fajar, F. Ciuandi, A. Munir, T. Informatika, and S. Kharisma Makassar, "Desain Aplikasi Daily Remainder Menggunakan Model-View Controller Dan Data Access Object Daily Remainder Application Design Using Model-View Controller and Data Access Object," 2023.

[22] E. Bautista-Villegas, "Metodologías agiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel," *Revista Amazonía Digital*, vol. 1, no. 1, p. e168, Jan. 2022, doi: 10.55873/rad.v1i1.168.

[23] V. Punitha, C. Mala, and Narendran Rajagopalan, "A novel deep learning model for detection of denial of service attacks in HTTP traffic over internet," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 33, no. 4, 2020.

[24] C.-O. Truică, E.-S. Apostol, J. Darmont, and T. B. Pedersen, "The Forgotten Document-Oriented Database Management Systems: An Overview and Benchmark of Native XML DODBMSes in Comparison with JSON DODBMSes," *Big Data Research*, Feb. 2021, doi: 10.1016/j.bdr.2021.100205.

[25] Z. Qu, X. Ling, T. Wang, X. Chen, S. Ji, and C. Wu, "AdvSQLi: Generating Adversarial SQL Injections against Real-world WAF-as-a-service," *IEEE Transactions on Information Forensics and Security*, Jan. 2024, doi: 10.1109/TIFS.2024.3350911.

[26] J. Hansen, T. Sutabri, U. Bina Darma Palembang, and H. Artikel, "Mendesain Cyber Security Untuk Mencegah Serangan DDoS Pada Website Menggunakan Metode Captcha," *Digital Transformation Technology (Digitech)* | *e*, vol. 3, no. 1, 2023, doi: 10.47709/digitech.v3i1.2764.