**Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)**
Vol.21. No.2, Agustus 2022, pp. 80~88
**P-ISSN : 1978-6603**
**E-ISSN : 2615-3475**


# Analisis Keamanan Untuk Mengetahui Vulnerability Pada DVWA Lab esting Menggunakan Penetration Testing Standart OWASP

**Ade Putra Armadhani[1], Dicky Nofriansyah[2], Khairi Ibnutama[3]**
[1,2,3] Sistem Informasi, STMIK Triguna Dharma

**Article Info**
**Article history:**
Received Aug 12th, 2022
Revised Aug 20th, 2022
Accepted Aug 30th, 2022

**Keyword:**
*Penetration Testing*
*OWASP*
*DVWA*
*Ethical Hacking*
*Web Security Analysis*
Keamanan *Website.*

**ABSTRACT**
Semakin hari teknologi semakin berkembang dan menuntut penggunanya untuk semakin berkembang mengikut teknologi terkini, teknologi semakin berkembang pastilah teknologi tersebut akan diperhatikan juga dari sisi keamanannya. Keamanan merupakan salah satu indikator penting dalam membangun sebuah aplikasi, mengingat akses ke internet menuju yang terbuka bebas bagi siapa saja. Sampai saat ini tidak ada aplikasi yang dapat dikatankan benar-benar aman. Sistem keamanan komputer dapat dikatakan sebuah cara yang dibuat untuk mengamankan fungsi, data, performa, atau proses yang ada pada sebuah komputer. Sebuah percobaan harus dilakukan untuk mengetahui apakah sebuah laman aman atau tidak dari aksi-aksi berbahaya yang dilakukan oleh penyerang. Salah satu cara untuk mengetahui apakah sistem kita aman atau tidak ialah dengan melakukan uji keamanan atau penetration testing. Atas dasar masalah tersebut, maka dengan memilih bidang cyber security penetration testing dengan mangadopsi metode standart Open Web Application Security Project (OWASP). Hadirnya OWASP sebagai standarisasi dan metode untuk uji keamanan sebuah sistem komputer berupaya untuk meningkatkan keamanan perangkat lunak dan didedikasikan untuk memungkinkan oraganisasi dalam mengembangkan, memperoleh, mengoperasikan, dan memelihara aplikasi terpercaya untuk menjamin keamanan yang dibuat atau dikembangkan. diharapkan dengan metode OWASP ini mampu menyelesaikan analisis dengan kriteria-kriteria yang sesuai standar OWASP secara transparan, tepat, efektif dan efisien. Hasil dari penelitian adalah melakukan analisis penetration testing standart OWASP dari celah keamanan yang sangat critical yaitu command injection & sql injection



**Corresponding Author:** *Ade Putra Armadhani
Nama : Ade Putra Armadhani
Program Studi : Sistem Informasi
STMIK Triguna Dharma
Email: mctavyish@gmail.com

## 1. PENDAHULUAN

Semakin hari teknologi semakin berkembang dan menuntut penggunanya untuk semakin berkembang mengikut teknologi terkini, teknologi semakin berkembang pastilah teknologi tersebut akan diperhatikan juga dari sisi keamanannya. Keamanan merupakan suatu usaha yang dilakukan untuk melindungi informasi yang terdapat didalamnya yang mengacu pada kerahasiaan[1].

Damn Vulnerable Web Application (DVWA) adalah aplikasi spesial untuk uji celah keamanan, berjalan menggunakan *service apache web server* yang berjalan pada protokol HTTP [2]. Tujuan utamanya adalah menjadi bantuan bagi para pemula dan profesional keamanan untuk menguji skill mereka dalam lingkungan hukum serta membantu web developer agar lebih memahami proses keamanan aplikasi web serta dapat membantu guru / murid / mahasiswa untuk mempelajari keamanan aplikasi web didalam kelas[3].

Vulnerability adalah suatu point kelemahan dimana suatu sistem rentan terhadap serangan [4]. Sistem keamanan komputer dapat dikatakan sebuah cara yang dibuat untuk mengamankan fungsi, data, performa, atau

proses yang ada pada sebuah komputer. Sebuah percobaan harus dilakukan untuk mengetahui apakah sebuah laman aman atau tidak dari aksi-aksi berbahaya yang dilakukan oleh penyerang. Salah satu cara untuk mengetahui apakah sistem kita aman atau tidak ialah dengan melakukan uji keamanan[5].

Pengujian penetrasi adalah serangkaian kegiatan yang dilakukan untuk mengidentifikasi dan mengeksploitasi kerentanan keamanan [6]. Pengujian penetrasi bukan hanya tentang menggunakan alat acak untuk memindai target untuk kerentanan, tetapi proses berorientasi detail yang melibatkan beberapa fase [7]. *Penetration testing* membantu mengkonfirmasi efektivitas atau ketidak efektifan langkah-langkah keamanan yang telah dilaksanakan, sehingga sangat membantu developer agar tidak memberikan code yang berbahaya atau yang berpotensi untuk disusupi..

## 2. METODE PENELITIAN

### 2.1 Tahapan Penelitian

Dalam melakukan penelitian, dilakukan metode penelitian Data Collecting (Teknik Pengumpulan Data) yaitu sebagai berikut:

a. Observasi
   Mengamati dan mencatat secara sistematik yang diselidiki, Data dikumpulkan dengan melihat secara langsung dari objek yang diteliti untuk mengetahui informasi dan kerentanan pada website DVWA LAB, objek yang diteliti merupakan lab uji keamanan DVWA..
b. Studi Literatur
   Literatur meliputi buku, artikel di jurnal dan buku digital berupa e-book. Studi pustaka yang dilakukan peniliti adalah pengumpulan data dari bahan-bahan referensi, arsip, jurnal, *e-book*, buku dan dokumen yang berhubungan dengan permasalahan dalam penelitian. Data diperoleh melalui studi kepustakaan (literature) yaitu dengan mencari bahan dari internet, jurnal dan perpustakaan serta buku yang sesuai dengan obyek yang akan diteliti.

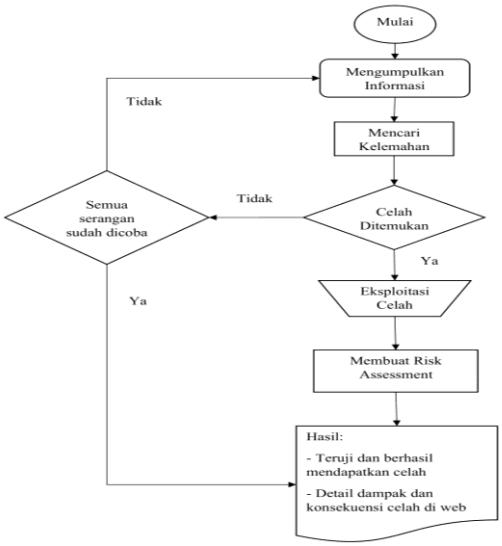
## 3. ANALISA DAN HASIL

### 3.1 Algoritma Sistem

Algoritma sistem merupakan penjelasan langkah-langkah dalam menyelesaikan masalah dalam Uji keamanan sistem pada website DVWA.

#### 3.1.1 Flowchart dari Algoritma Penyelesaian OWASP

Flowchart Algoritma yang dirancang untuk menentukan skala prioritas Web *Penetration testing* DVWA Lab Testing dengan menggunakan metode OWASP berdasarkan kriteria-kriteria yang telah ditentukan berikut[8]:




Gambar 1: Flowchart algoritma OWASP

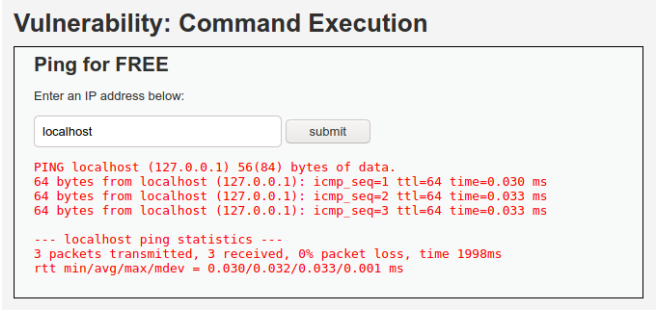
1. Penetration testing dimulai dengan mengumpulkan semua kemungkinan informasi yang tersedia mengenai infrastruktur dan aplikasi yang terlibat. Tahap ini sangat penting karena tanpa pemahaman yang kuat tentang teknologi yang mendasari yang terlibat, bagian mungkin terlewatkan selama fase pengujian[9].

2. Mencari kelemahan merupakan hal yang penting dalam penetration testing sehingga output dari uji keamanan didapatkan, maka dari itu tahap pertama harus mengumpulkan informasi sebanyaknya sehingga memudahkan penguji dalam merangkum dimana letak celah keamanan sebelum menemukan celah keamanan.
3. Jika menemukan celah keamanan maka bisa melanjutkan tahap selanjutnya, jika tidak maka harus lebih berusaha lagi untuk mencari celah keamanan dengan mencoba semua cara dan jika tidak menemukan celah keamanan maka harus mencari informasi sebanyaknya lagi.
4. Penguji harus mencoba mengeksploitasi semua kerentanan yang ditemukan. Jika tidak maka akan mencoba semua cara untuk mencari celah keamanan sehingga apabila tidak ditemukan celah keamanan maka sudah selesai tugas penguji dan membuat laporan bahwa tidak menemukan celah.
5. Risk Assessment proses untuk melihat dampak dari terjadinya celah keamanan pada sistem tersebut dan melakukan penilaian celah biasanya berupa low, medium, high dan critical[10]. Risk Assessment tidaklah asal-asal melainkan harus mengikuti CVSS / Score pembuatan *risk vulnerability assessment*.
6. Tahap terakhir ialah menulis laporan / temuan celah keamanan dalam format yang sudah ditentukan lengkap dengan cara menemukan kelemahan sistem tersebut.

## 3.2. Pengujian

### 3.2.1 Vulnerability Command Execution

Dalam fitur "*Ping for FREE*" penguji dalam melakukan ping layaknya menggunakan terminal / CMD sehingga *user* dapat mengetahui kecepatan *response* dari IP dituju.




Gambar 2 Uji keamanan *Command execution*

Terlihat pada gambar dibawah bahwa *function* shell_exec() digunakan untuk melakukan *request* ping yang mana melakukan ping localhost menggunakan terminal pada server sehingga penyerang bisa saja menggunakan perintah Linux pada fitur ini.

```php
<?php

if( isset( $_POST[ 'submit' ] ) ) {

    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if (stristr(php_uname('s'), 'Windows NT')) {

        $cmd = shell_exec( 'ping  ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    } else {

        $cmd = shell_exec( 'ping  -c 3 ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    }

}
?>
```

Gambar 3 *Source code low security*

Jika menambahkan (&&) pada fitur ini yang mana perintah dari (&&) pada *Linux* yaitu melanjutkan perintah *Linux* selanjutnya. Lalu menambahkan (ls -la) yaitu perintah linux untuk menampilkan file atau *directory* pada *server*.

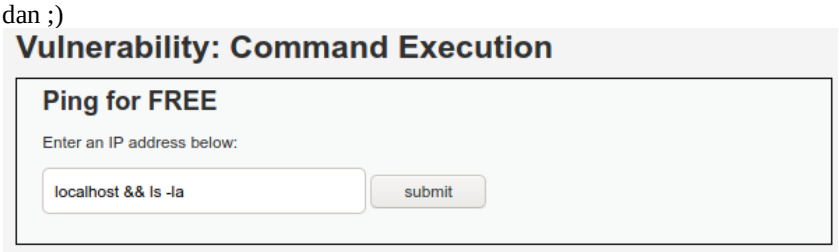

Gambar 4. Uji keamanan *command execution* bagian 2

Untuk mencegah celah seperti ini kita bisa membatasi perintah (&& atau ;) ini kita bisa masukan ke *array* sebagai *blacklist*.

```php
<?php

if( isset( $_POST[ 'submit' ] ) ) {

    $target = $_REQUEST[ 'ip' ];

    // Remove any of the charactars in the array (blacklist).
    $substitutions = array(
        '&&' => '',
        ';' => '',
    );

    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if (stristr(php_uname('s'), 'Windows NT')) {

        $cmd = shell_exec( 'ping  ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    } else {

        $cmd = shell_exec( 'ping  -c 3 ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    }
}

?>
```

Gambar 5. *Source code medium security*

Pada gambar dibawah tidak bisa melalakukan *command execution* lagi karena ada perintah yang di *blacklist* yaitu (&& dan ;)




Gambar 6. Uji keamanan *command execution* bagian 3

Walaupun telah ditambahkan *word blacklist* tapi tetap saja ini mempunyai kelemahan yang mana harus memasukan perintah perintah tertentu yang ingin di *blacklist* padahal *function* yang mirip seperti (&& dan ;) masih ada yaitu menggunakan *pipe line* ( | ). Terlihat pada gambar berikut ini dapat melakukan celah *command execution* lagi.


Gambar 7. Uji keamanan *command execution* bagian 4

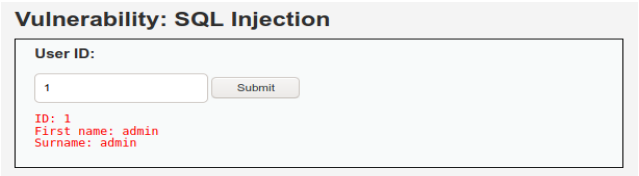
Untuk menutup celah ini *developer* bisa mengikuti atau menambahkan *source code* pada gambar dibawah ini sehingga penyerang tidak dapat melakukan *command execution* lagi terlampir pada gambar berikut.

```php
<?php

if( isset( $_POST[ 'submit' ] ) ) {

    $target = $_REQUEST["ip"];

    $target = stripslashes( $target );


    // Split the IP into 4 octects
    $octet = explode(".", $target);

    // Check IF each octet is an integer
    if ((is_numeric($octet[0])) && (is_numeric($octet[1])) && (is_numeric($octet[2])) && (is_numeric($octet[3])) && (sizeof($octet) == 4)  ) {

    // If all 4 octets are int's put the IP back together.
    $target = $octet[0].'.'.$octet[1].'.'.$octet[2].'.'.$octet[3];


        // Determine OS and execute the ping command.
        if (stristr(php_uname('s'), 'Windows NT')) {

            $cmd = shell_exec( 'ping  ' . $target );
            echo '<pre>'.$cmd.'</pre>';

        } else {

            $cmd = shell_exec( 'ping  -c 3 ' . $target );
            echo '<pre>'.$cmd.'</pre>';

        }

    }

    else {
        echo '<pre>ERROR: You have entered an invalid IP</pre>';
    }


}

?>
```

Gambar 8. *Source code high security*

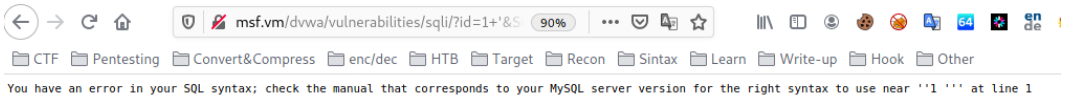
### 3.2.2 Vulnerability SQL Injection

Pada celah keamanan SQL *Injection* penyerang dapat menyusupi *server database* yakni MySQL bahkan celah keamanan ini masih ada pada saat ini.




Gambar 9. Uji keamanan SQL *Injection* bagian 1

Pada umumnya celah keamanan SQL *Injection* dapat dipicu menggunakan petik satu (‘) yakni merupakan *query* dari MySQL sehingga kita bisa memasuki *query* MySQL yang kita inginkan bila mana *query* MySQL valid.



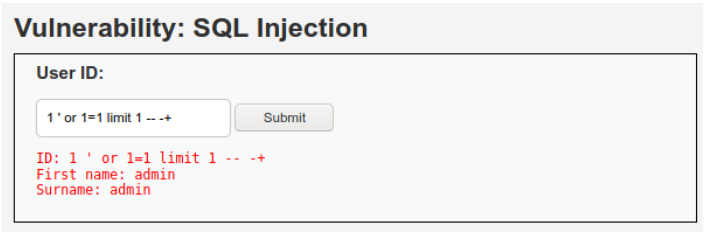

Gambar 10. Uji keamanan SQL *Injection* bagian 2

Pada gambar diatas terlihat memasukan *query* yang tidak valid maka dari itu *developer* harus memasukan *query* yang tepat.

```php
<?php
if(isset($_GET['Submit'])){
    // Retrieve data
    $id = $_GET['id'];
    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );
    $num = mysql_numrows($result);
    $i = 0;
    while ($i < $num) {
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");
        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';
        $i++;
    }
}
?>
```

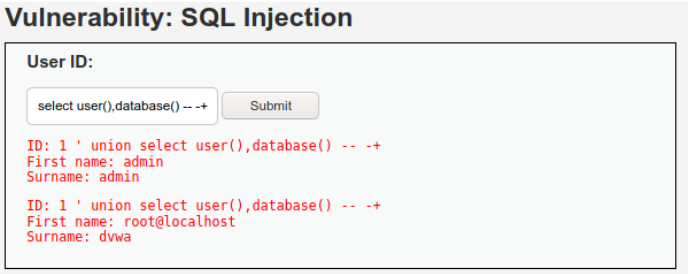
Gambar 11. *Source code low security* SQL *Injection*

Saat ini menggunakan *query* pemanggilan yaitu SELECT atau memilih first_name dan last_name dari table users yang mana mencari menggunakan id. Bilamana menambahkan petik satu (‘) maka query akan menjadi seperti ini : "SELECT first_name, last_name FROM users WHERE user_id = '$id' ‘" untuk mencoba *query* yang valid untuk MySQL mari menggunakan *query bolean* pada MySQL 1=1 dan hasilnya *true*, seperti ini : "SELECT first_name, last_name FROM users WHERE user_id = '$id' ‘ or 1=1 limit 1 -- -+"




Gambar 12. Uji keamanan SQL *Injection* bagian 3

Jika kita memasukan *query* yang valid maka kita bisa membaca nama *database* dan *user* pada MySQL DVWA lab testing seperti gambar dibawah ini.



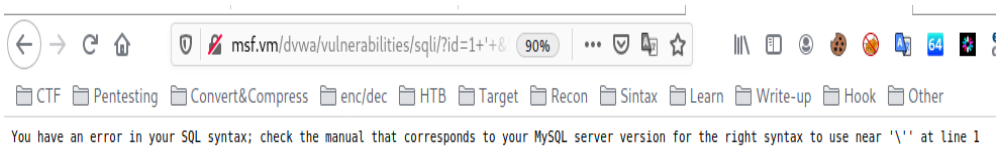

Gambar 13. Uji keamanan SQL *Injection* bagian 4

Untuk menutup celah keamanan ini bisa menggunakan *function* yang sering digunakan yaitu mysql_real_escape_string($id); seperti gambar dibawah

```php
<?php

if (isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];
    $id = mysql_real_escape_string($id);

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = $id";

    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

    $num = mysql_numrows($result);

    $i=0;

    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
?>
```

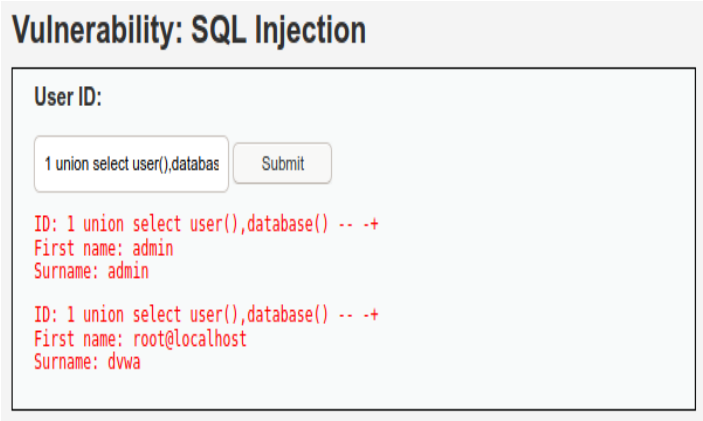
Gambar 14. *Source code mediumsecurity* SQL *Injection*

Jika menambahkan petik satu (‘) sebagai pemicu *query* MySQL maka pada kali ini mendapatkan error yang beda dari yang sebelumnya, seperti ini :
“You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\'' at line 1”




Gambar 15. Tampilan error MySQL

Sayangnya *function* tersebut masih bisa di *bypass* dengan cara tidak menambahkan petik satu (‘) dan melanjutkan eksekusi *query* yang valid seperti *query* ini : “SELECT first_name, last_name FROM users WHERE user_id = 1 union select user(),database() -- -+;”




Gambar 16. Uji keamanan SQL *Injection* bagian 5

Ada penambahan *function* PHP jika ingin menutup celah ini yaitu menggunakan *function* stripslashes(). Karakter *backslash* (“\”) dalam bahasa pemrograman dikenal sebagai karakter *escape*, yaitu karakter yang digunakan untuk memungkinkan karakter khusus ditampilkan bersama dengan karakter biasa. Salah satu contoh dari karakter khusus adalah karakter *apostrophe* (tanda petik, kutip tunggal). Di dalam program PHP, *apostrophe* digunakan untuk tanda yang mengapit data yang dianggap sebagai karakter atau *string*.

```php
<?php

if (isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];
    $id = stripslashes($id);
    $id = mysql_real_escape_string($id);

    if (is_numeric($id)){

        $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
        $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

        $num = mysql_numrows($result);

        $i=0;

        while ($i < $num) {

            $first = mysql_result($result,$i,"first_name");
            $last = mysql_result($result,$i,"last_name");

            echo '<pre>';
            echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
            echo '</pre>';

            $i++;
        }
    }
}
?>
```

Gambar 17. *Source code high security* SQL *Injection*

## 4. KESIMPULAN

Berdasarkan hasil analisa dan pengujian, metode Standart OWASP dapat diterapkan dalam hal *penetration testing* pemecahan masalah uji keamanan suatu *website* dan mengikuti standarisasi OWASP dalam hal uji keamanan. Pada hasil pengujian, terdapat *function* dan *query* MySQL yang tidak di *filter* maka dalam hal ini tidak layak pakai dalam hal *website* karna penyerang dapat mengambil alih *server* dan *database*.

## UCAPAN TERIMA KASIH

Pada kesempatan ini penulis mengucapkan banyak terimakasih kepada kedua orang tua yang telah banyak memberikan dukungan moril dan materil, tidak terkecuali doa yang senantiasa dipanjatkan sehingga penulis dapat menyelesaikan penelitian ini.

Penyusunan jurnal ini juga tidak terlepas dari bantuan berbagai pihak. Oleh karena itu dengan segala kerendahan hati, diucapkan terimakasih yang sebesar-besarnya kepada: Bapak Dr. Dicky Nofriansyah, S.Kom., M.Kom dan Khairi Ibnutama, S.Kom., M.Kom yang telah banyak membantu baik moril dan pengetahuan yang telah di share.

## REFERENSI

[1] Y. W, I. Riadi, dan A. Yudhana, "Analisis Deteksi Vulnerability Pada Web Server Open Journal System Menggunakan OWASP Scanner," *Jurnal Rekayasa Teknologi Informasi (JURTI)*, vol. 2, no. 1. hal. 1, 2018, doi: 10.30872/jurti.v2i1.1319.

[2] I. Kamilah dan A. Hendri Hendrawan, "Analisis Keamanan Vulnerability pada Server Absensi Kehadiran Laboratorium di Program Studi Teknik Informatika," *Pros. Semnastek*, vol. 16, no. 0, hal. 1–9, 2019, [Daring]. Tersedia pada: https://jurnal.umj.ac.id/index.php/semnastek/article/view/5233.

[3] D. S. Firdaus dan A. H. Hendrawan, "Analisis Keamanan Vulnerability pada Server Cloud Open Media Vault di Fakultas Teknik Universitas Ibn Khaldun Bogor," *Anal. Keamanan Vulnerability pada Serv. Cloud Open Media Vault di Fak. Tek. Univ. Ibn Khaldun Bogor*, hal. 1–9, 2019, [Daring]. Tersedia pada: https://jurnal.umj.ac.id/index.php/semnastek/article/view/5241.

[4] A. Susanto dan W. K. Raharja, "Simulation and Analysis of Network Security Performance Using Attack Vector Method for Public Wifi Communication," *IJICS (International J. Informatics Comput. Sci.*, vol. 5, no. 1, hal. 7–15, 2021, doi: 10.30865/ijics.v5i1.2764.

[5] S. Sahren, R. A. Dalimuthe, dan M. Amin, "Penetration Testing Untuk Deteksi Vulnerability Sistem Informasi Kampus," *Pros. Semin. Nas. Ris. Inf. Sci.*, vol. 1, no. September, hal. 994, 2019, doi: 10.30645/senaris.v1i0.109.

[6] Muhammad Athallariq Rabbani, Avon Budiyono, dan Adityas Widjajarto, "Implementasi dan Analisis Security Auditing Menggunakan Open Source Software Dengan Framework Mitre ATT&CK," *e-Proceeding Eng.*, vol. 7, no. 2, hal. 7080–7087, 2020.

[7] D. P. Utomo dan M. Mesran, "Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung," *J. Media Inform. Budidarma*, vol. 4, no. 2, hal. 437, 2020, doi:

10.30865/mib.v4i2.2080.
[8] M. S. Mohammad Suryawinata, *Buku Ajar Mata Kuliah Pengembangan Aplikasi Berbasis Web*. 2019.
[9] S. Hidayatulloh dan D. Saptadiaji, "Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP)," *J. Algoritm.*, vol. 18, no. 1, hal. 77–86, 2021, doi: 10.33364/algoritma/v.18-1.827.
[10] K. E. Putra, "Pengukuran Resiko Jaringan Komputer Menggunakan Technical Risk Assessment pada SMK Muhammadiyah 2 Pekanbaru," hal. 18–19, 2017.

**BIBLIOGRAFI PENULIS**

| | | |
|---|---|---|
| Nama | : | Ade Putra Armadhani, Amd.Kom |
| Nirm | : | 2019020781 |
| Tanggal Lahir | : | 30 Mei 1998 |
| Jenis Kelamin | : | Laki-Laki |
| Jenjang Pendidikan | : | SMK Yapim Batang Kuis<br>D3 STMIK Trigunadharma |
| Bidang Ilmu | : | Cyber Security, Web Devloper, System Administrar, Linux Management, Capture The Flag, Digital Forensic |
| Deskripsi | : | Mahasiswa stambuk 2016 yang aktif di UKM Mahasiswa yaitu Cyber LDK Irmastri |
| Prestasi | : | Finalis Born To Protect Kominfo di PUSTIKNAS Kominfo.<br>Finalis Gemastik12 di Universitas Telkom Bandung.<br>Finalis IndoSecurity di Jakarta Conventional Center.<br>Finalis KSSI TNI-AD di Jakarta.<br>Narasumber Penguji keamanan oleh BSSN di Makassar. |

| | | |
|---|---|---|
| Nama | : | Dr. Dicky Nofriansyah, S.Kom., M.Kom |
| Tanggal Lahir | : | 31 Oktober 1989 |
| NIDN | : | 0131058901 |
| Jabatan | : | Dosen |
| Email | : | dickynofriansyah@ymail.com |
| Jenis Kelamin | : | Laki-Laki |
| Jenjang Pendidikan | : | S1 Universitas Budidarma Medan<br>S2 Universitas Putra Indonesia Yptk Padang<br>S3 Universitas Negeri Padang |
| Bidang Ilmu | : | Data Mining, Artificial Intelligence, Computer Security, Sistem Pendukung Keputusan, Pemograman VB.NET 2008 dan Teknik Riset Operasional |
| Deskripsi | : | Dosen Tetap STMIK Triguna Dharma pada program studi Sistem Informasi yang aktif mengajar dan fokus pada bidang Data Mining, Artificial Intelligence, Computer Security, Sistem Pendukung Keputusan, Pemograman VB.NET 2008 dan Teknik Riset Operasional. |

| | | |
|---|---|---|
| Nama | : | Khairi Ibnutama, S.Kom., M.Kom |
| Tanggal Lahir | : | 24 Juni 1987 |
| NIDN | : | 0124068702 |
| Jabatan | : | Dosen |
| Jenis Kelamin | : | Laki-laki |
| Jenjang Pendidikan | : | D3 STMIK Triguna Dharma Medan<br>S1 STMIK Triguna Dharma Medan<br>S2 Universitas Putra Indonesia Yptk Padang |
| Jabatan | : | Dosen |
| Email | : | mr.ibnutama@gmail.com |
| Bidang Ilmu | : | Website, Android/iOS, Disain Grafis, IOT |
| Deskripsi | : | Memulai karir di dunia IT sejak tahun 2007 sebagai freelancer pada sebagian besar bidang pemrograman komputer. Pernah menjadi pemateri pada Pelatihan Operator Sub Domain Website OPD Pemerintah Kota Medan yang diadakan oleh Dinas Komunikasi dan Informatika Kota Medan. Aktif dalam Tridharma Perguruan Tinggisebagai Dosen tetap di STMIK Triguna Dharma, serta menjabat sebagai Tim Ahli pada Yayasan Kesejahteraan Anak Pesisir Indonesia (YKAPI) |