ANALISIS KEAMANAN APLIKASI DATA POKOK PENDIDIKAN (DAPODIK) MENGGUNAKAN PENETRATION TESTING DAN SQL INJECTION

Ade Bastian[1], Harun Sujadi[2], Latiful Abror[3]
[1,,2,3]Program, Studi Informatika Fakultas Teknik Universitas Majalengka
Email: [1]adebastian@unma.ac.id, [2]harunsujadi@unma.ac.id, [3]latifulabror@gmail.com

ABSTRAK
Seringkali masalah keamanan aplikasi terabaikan justru setelah semua peralatan dan infrastruktur pengaman telah terpasang. Bahkan pentingnya pengamanan baru disadari setelah terjadi bencana. Kerugian sebuah institusi/organisasi yang diakibatkan dari sebuah serangan terhadap sistem aplikasi sangatlah besar, tetapi hal ini sangat sukar dideteksi, karena secara umum tidak akan diakui dengan berbagai alasan. Tanpa pengamanan sistem aplikasi yang baik, penerapan teknologi sehebat apapun akan sangat membahayakan institusi/organisasi itu sendiri. Nilai informasi yang begitu penting dan strategis mengakibatkan serangan dan ancaman terhadap aplikasi dan arus informasi semakin meningkat. Kebutuhan keamanan sistem aplikasi timbul dari kebutuhan untuk melindungi data. Pertama, dari kehilangan dan kerusakan data. Kedua, adanya pihak yang tidak diijinkan hendak mengakses atau mengubah data. Rancangan arsitektur Aplikasi Dapodik yang dihasilkan dibatasi yaitu hanya pada aplikasi Dapodik berbasis web yang diterapkan secara offline dan online yang terkoneksi dengan internet. Pengujian keamanan aplikasi Dapodik menggunakan penetration testing dan SQL Injection. Berdasarkan hasil pengujian keamanan, dapat dinyatakan bahwa tingkat ancaman terhadap web server dan aplikasi Dapodik berada pada level aman. Hal tersebut menunjukkan bahwa tidak ada celah yang memungkinkan terjadinya ancaman dan akses ilegal yang berpotensi merusak sistem.

Kata Kunci: Aplikasi Dapodik, Keamanan Aplikasi, Penetration Testing, SQL Injection

1. PENDAHULUAN

1.1. Latar Belakang

Berisi Teknologi zaman sekarang yang semakin berkembang salah satunya aplikasi berbasis web. Saat ini banyak sekali aplikasi web yang sudah dibuat dan di kembangkan seperti contohnya jejaring sosial, *e-learning*, termasuk data pokok pendidikan yang berbasis web. Dengan diterbitkannya Instruksi Menteri Pendidikan Nasional nomor 2 tahun 2011, Sekretariat Direktorat Jenderal Pendidikan Menengah memiliki kewenangan tambahan untuk merancang prosedur pengumpulan data, melakukan sosialisasi dan membangun sistem pengumpulan dan penyimpanan data yang cepat dan efisien serta mengkoordinir pengumpulan semua data pokok pendidikan dari satuan pendidikan/sekolah yang berada dibawah pembinaan masing-masing Direktorat Jenderal. Pengumpulan data pokok pendidikan yang dimaksud yaitu meliputi pengumpulan, pengolahan, dan penyajian data.

Perancangan arsitektur sistem aplikasi itu proses perencanaan yang harus dikelola berdasarkan suatu petunjuk yang jelas dengan tujuan menyelaraskan strategi sistem dan strategi teknologi untuk memberikan hasil yang maksimal bagi lembaga. Saat ini belum terdapat kerangka dasar yang khusus untuk melakukan perancangan arsitektur teknologi informasi untuk institusi pendidikan.

Untuk melaksanakan perencanaan pendidikan, maupun untuk melaksanaan program-program pendidikan secara tepat sasaran, dibutuhkan data yang cepat, lengkap, valid, akuntabel dan terus *up to date*. Dengan ketersediaan data yang cepat, lengkap, valid, akuntabel dan *up to date* tersebut, maka proses perencanaan, pelaksanaan, pelaporan dan evaluasi kinerja program-program pendidikan nasional dapat dilaksanakan dengan lebih terukur, tepat sasaran, efektif, efisien dan berkelanjutan.

Serangan injeksi terjadi ketika seseorang yang tidak sah, mengirimkan perintah SQL berbahaya ke server yang cukup sering terjadi karena hampir semua aplikasi modern menggunakan *database* terpusat untuk menyampaikan informasi. (Shar, L.K., 2013: 66-67). Jalur yang paling banyak diserang adalah data input yang tidak divalidasikan. (Singh, P., 2015: 16-21).

SQL injeksi adalah beberapa perintah SQL berbahaya dapat dikirimkan ke SQL. SQL injeksi sudah masuk sepuluh besar resiko aplikasi web seperti pada tabel dibawah ini :

Tabel 1. Top 10 Resiko Aplikasi Web

No	*OWASP Top 10-2017*
1	*A1:2017-Injection*
2	*A2:2017-Broken Authentication*
3	*A3:2017:Sensitive Data Exposure*
4	*A4:2017=XML External Entities (XXE)*
5	*A5:2017-Broken Access Control*
6	*A6:2017-Security Misconfiguration*
7	*A7:2017-Cross-Site Scripting (XSS)*
8	*A8:2017-Insecure Deserialization*
9	*A9:2017-Using Components with Known Vulnerabilities*
10	*A10:2017-Insufficient Logging and Monitoring*

Pengujian penetrasi merupakan salah satu metode untuk menguji kelengkapan, keterpaduan, operasional dan dasar dunia komputer yang terdiri dari perangkat keras, perangkat lunak dan manusia. Proses dan penanggulanganya melibatkan analisa

sistem, kekurangan atau konfigurasi sistem, dan kelemahan *software*. (Wahyudi, 2019: 6). Pengujian penetrasi juga disebut *pen testing* atau *ethical hacking* adalah praktik pengujian sistem komputer, jaringan, atau web aplikasi untuk menemukan kerentanan keamanan yang dapat dieksploitasi oleh penyerang. Pengujian penetrasi dapat diotomatiskan dengan perangkat lunak aplikasi atau dilakukan secara manual. (Kesharwani, 2018: 193-200)

1.2. Tinjuan Pustaka

1.2.1 SQL Injection

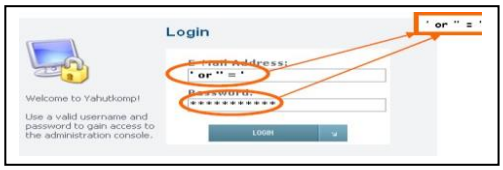
SQL Injection adalah sebuah aksi hacking yang dilakukan di aplikasi client dengan cara memodifikasi perintah *SQL* yang ada di memori aplikasi *client*, *Sql Injection* merupakan teknik mengekploitasi aplikasi berbasis yang di dalamnya menggunakan basis data untuk penyimpanan data.
Penyebab utama terjadinya *SQL Injection* adalah Tidak adanya penanganan terhadap karakter – karakter tanda petik satu ‟ dan juga karakter double minus -- yang menyebabkan suatu aplikasi dapat disisipi dengan perintah SQL, Sehingga seorang Hacker menyisipkan perintah SQL kedalam suatu parameter maupun suatu form. *Bug SQL Injection* sangat berbahaya, karena Teknik ini memungkinkan seseorang dapat login kedalam sistem tanpa harus memiliki account, Selain itu *SQL injection* juga memungkinkan seseorang merubah, menghapus, maupun menambahkan data–data yang berada didalam database. Bahkan yang lebih berbahaya lagi yaitu mematikan database itu sendiri, sehingga tidak bisa memberi layanan kepada *web server*. (Yudistira, 2012). Sebagai contoh misalkan ada aplikasi berbasis *web* dengan *source code* seperti ini:

```
$SQL = "select * from login where
username ="$username" and
password =„$password"";
```

Dengan metode pengiriman data *GET* ataupun *POST* maka *query* seperti ini dapat di injeksi dengan mengisikan *string* „*or*'""=‟ pada login *form* di bawah ini :

```
$SQL = "select * from login where
username = "$username" and
password="pass" or „='"";
```

Dengan sintak *SQL Injection* ini hasil selection akan selalu True sehingga aplikasi tersebut dapat di tembus sebagai contoh di bawah ini :



Gambar 1. SQL Injection pada Login Form

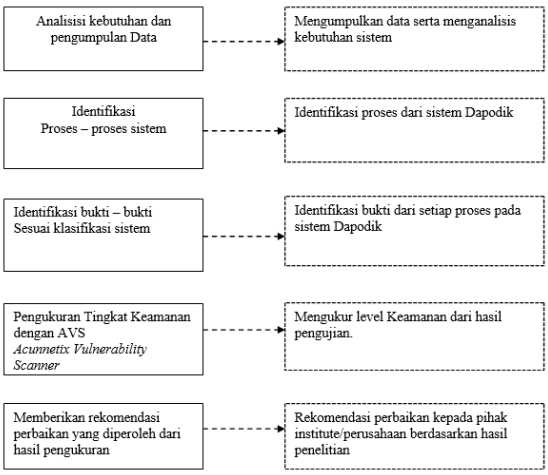
1.2.2 Acunetix Web Vulnerability Scanner

Acunetix web vulnerability Scanner digunakan untuk mngetahui celah suatu Website seperti *SQL Injection, Cross Site Scripting, Upload files, XSS attack,* dll. Dan software ini dapat menguji keamanan suatu *website* secara otomatis dengan mengaudit *website* dengan memeriksa kerentanan *hacking* untuk dieksploitasi.
Acunetix Web Vulnerability Scanner (WVS), secara otomatis menganalisa aplikasi web Anda dan menemukan menelusuri situs Anda , *SQL injection* berbahaya , *scripting Cross- Site* dan kerentanan lain yang mengekspos bisnis online Anda . laporan ringkas mengidentifikasi di mana aplikasi web perlu diperbaiki , sehingga memungkinkan Anda untuk melindungi bisnis Anda dari yang akan datang serangan *hacker*. *Acunetix* telah memelopori aplikasi *web scanning* keamanan dan telah membentuk memimpin rekayasa dalam analisis situs dan deteksi kerentanan dengan fitur-fitur inovatif. (Sutanta, 2008)
Metode kualitatif dengan menggunakan beberapa *tools* berupa perangkat lunak dan cara-cara tertentu yang lazim digunakan untuk menguji keamanan aplikasi. Tahap-tahap yang dilakukan adalah sebagai berikut :

1. Tahap Inisiasi, pada tahap ini dilakukan penelusuran dan pengkajian *literatur-literatur* yang berhubungan dengan keamanan aplikasi.
2. Tahap Investigasi, pada tahap ini dilakukan penyelidikan terhadap *web server*, program aplikasi yang digunakan.
3. Tahap Pengujian, pada tahap ini dilakukan pengujian terhadap keamanan aplikasi dengan menggunakan *tools*, yaitu *Acunetix web vurnerability scanner* dengan metode yang lazim digunakan dalam pengujian keamanan aplikasi dan sistem.
4. Tahap Verifikasi, pada tahap ini dilakukan verifikasi terhadap keamanan aplikasi untuk pemberitahuan kepada admin untuk dilakukan perbaikan-perbaikan atas dasar hasil investigasi dan pengujian pada aspek pemrograman.

1.3. Metodologi Penelitian

Untuk melakukan analisis dan pengujian keamanan aplikasi Dapodik, disusun tahapan seperti tergambar pada Gambar 1 dibawah ini :

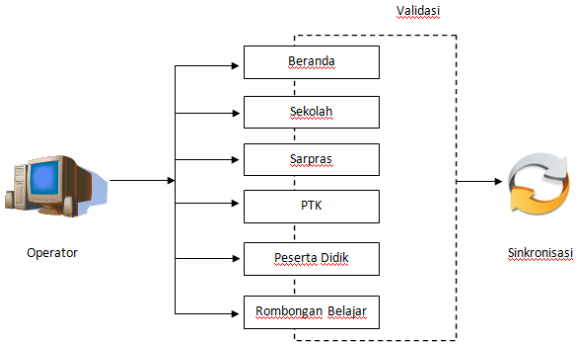


Gambar 2. Tahapan Penelitian

Studi lapangan ditempuh untuk pengumpulan data dengan cara bertanya langsung kepada Guru atau pihak sekolah mengenai mekanisme/ prosedur pengelolaan aplikasi data pokok Pendidikan pendataan sekolah khususnya operator sekolah atau staf yang menjalankan aplikasi Dapodik tersebut.

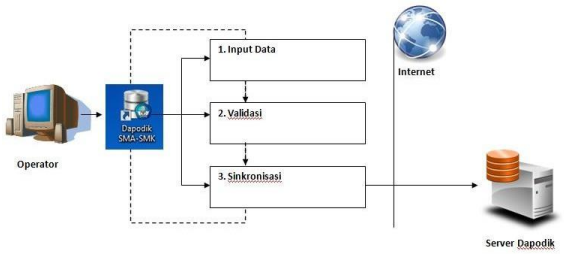
2. PEMBAHASAN

Aplikasi Dapodik meliputi 3 proses yaitu operator melakukan penginputan data diantaranya data sekolahm PTK, peserta didik, sarpras dan lainnya. Setelah itu divalidasi untuk memverifikasi data yang valid, dan kemudian menghubungkan ke internet untuk proses pengiriman data/ sinkronisasi ke server Dapodik pusat. Arsitektur Aplikasi Dapodik dapat dilihat pada Gambar 2 dibawah ini :



Gambar 3. Arsitektur Aplikasi Dapodik

Untuk menghasilkan data yang alid, operator melakukan penginputan data secara lengkap meliputi data beranda, data sekolah, data sarpras, data PTK, data peserta didik dan data rombel. Setelah semua data divalidasi dan semua data valid kemudian disingkronisasi yaitu pengiriman data valid yang sudah diinput ke server Dapodik pusat. Arsitektur data aplikasi Dapodik dapat dilihat pada Gambar 3 dibawah ini :



Gambar 4. Arsitektur Data Aplikasi Dapodik

2.1. Injeksi di URL

Panjang Sebelum tahap pengujian pada umumnya sebuah aplikasi web yang kemungkinan memiliki vulnerability, adalah seperti ini sebagai contoh `http://www.dapodik.com/contoh.php?id=123` pada alamat tersebut halaman web memiliki variable "id" dan memiliki nilai (value) "123". Variable – variable inilah yang membuat web dapat di eksplorasi. Contoh halaman yang semula adalah seperti ini :

`http://localhost:8090`

Kemudian setelah dimodifikasi seperti ini :

`http://localhost:8090''`

Kemudian muncul pesan seperti ini :

Gambar 5. Pesan Error Penambahan Tanda Petik

Artinya web tersebut *vulnerability* sehingga dugaan awal adalah aplikasi Dapodik memiliki *vulnerability* pada URL nya. Langkah bertikutnya adalah menghitung tabel basis data yang ada pada aplikasi Dapodik. Cara yang digunakan memodifikasi alamat URL dengan tambahan perintah *Order By*. Halaman semua seperti ini :

`http://localhost:8090`

Kemudian setelah dimodifikasi L

`http://localhost:8090+order+by+1`

Kemudian muncuk pesan seperti ini :

Gambar 6. Pesan Error Perintah Order By

Hal ini menandakan aplikasi Dapodik tidak memiliki tabel basis data. Alamat tersebut tidak memiliki operasi basis data, sehingga dengan kata lain URL dari aplikasi Dapodik tidak dapat diinjeksi karena sudah aman.

2.2. Analisis Method

Metode atau *method* adalah suatu proses bagaimana suatu data dikirimkan ke web server. Pada pemrograman web dikenal 2 *method* yaitu *method POST* dan *method GET*. Keduanya memiliki perbedaan, dimana perbedaan yang paling mencolok adalah variable yang dikirimkan. Didalam *method GET*, variable yang dikirimkan terlihat di URL, sedangkan didalam *method POST* variable yang dikirimkan tersembunyi.

Contoh alamat URL menggunakan *method GET* :
`http://www.dapodik.com/contoh.php?id=123`

Contoh alamat URL menggunakan *method POST* :
`http://www.dapodik.com/contoh.php`

Berdasarkan contoh di atas dapat dilihat bahwa pada method GET di URL nya terdapat variabel "id" dengan value bernilai "123". Variabel inilah yang dapat di modifikasi *client* untuk menginjeksi web tersebut. Berbeda dengan method POST, method ini menyembunyikan variable data yang akan di kirimkan ke web server, contohnya adalah pada aplikasi web Dapodik. Pada saat web tersebut di buka alamat URL nya adalah `http://www.dapodik.com/contoh.php`.
Untuk melihat perbedaan *method GET* dan *POST* lebih rinci dapat dilihat di perbedaan berikut :

1. *Method GET*
   - Variable dapat di lihat di URL, data setiap variable di pisahkan dengan & contoh `http://url/page.php?get1=nilai2&get2=nilai2`
   - Dibatasi oleh panjang string sebanyak 2047 karakter.
   - Memungkinkan pengunjung langsung memasukan nilai variabel pada form proses.
2. *Method POST*
   - Nilai variabel tersembunyi untuk pengunjung.
   - Lebih aman karena lebih susah di mainkan oleh pengunjung melalui pergantian nama variabel.
   - Tidak di batasi oleh panjang string.

2.3. Pengujian Keamanan Dapodik

Web server Aplikasi Dapodik di bangun menggunakan *Apache 2.4.6* dengan sistem operasi windows, bahasa pemrograman *PHP* versi 5.4.19 dan database aplikasi nya yang berekstensi (*.prf*). yaitu profil pengguna yang berisi preferensi dan informasi pengguna seperti tanda tangan, pengaturan akun email dan pengaturan lainnya.
Metode yang digunakan dalam penelitian ini adalah metode kualitatif dengan menggunakan beberapa *tools* berupa perangkat lunak dan cara-cara tertentu yang lazim digunakan untuk menguji keamanan aplikasi. Tahap-tahap yang dilakukan adalah sebagai berikut :

a. Tahap Inisiasi, pada tahap ini dilakukan penelusuran dan pengkajian literatur- literatur yang berhubungan dengan keamanan aplikasi.
b. Tahap Investigasi, pada tahap ini dilakukan penyelidikan terhadap *web server*, program aplikasi, dan *database server* yang digunakan.
c. Tahap Pengujian, pada tahap ini dilakukan pengujian terhadap keamanan aplikasi dengan menggunakan dua tools, yaitu *Acunetix Web Vulnerability Scanner* versi 6.0 (untuk menguji *web server* dan program aplikasi) dan *Shadow Database Scanner* versi 7.75 (untuk menguji *database server*) dengan metode yang lazim digunakan dalam pengujian keamanan aplikasi dan sistem.
d. Tahap Verifikasi, pada tahap ini dilakukanverifikasi terhadap keamanan aplikasi setelah dilakukan perbaikan-perbaikan atas dasar hasil investigasi dan pengujian pada aspek pemrograman maupun konfigurasi database server yang digunakan untuk memastikan bahwa database tersebut siap diterapkan untuk aplikasi Dapodik.

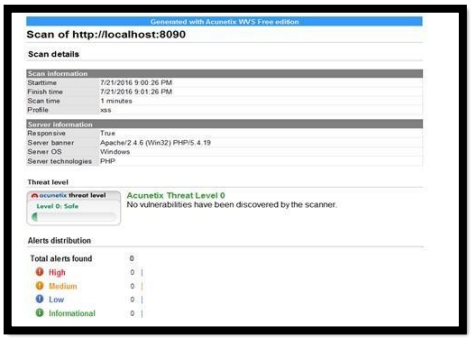
Aspek-aspek keamanan aplikasi yang diteliti meliputi :

a. *Web Server*
b. *Program aplikasi*
c. *Database Server*

Untuk melakukan analisis keamanan aplpikasi Dapodik, dilakukan menggunakan 3 (tiga) *software* yaitu :

a. *Accunnetix Web Vulneability Scanner* versi 6.5
b. *Shadow Database Scanner* versi 7.75
c. *Havij* versi 1.16

Hasil analisis yang diperoleh menggunakan *Accunetix Web Vulnerability Scanner* adalah sebagai berikut :



Gambar 7. Hasil Analisis Web Server

Berdasarkan hasil analisis di atas dapat diketahui bahwa tingkat ancaman terhadap web server Aplikasi Dapodik berada pada level 0. Artinya aplikasi web Dapodik ini Aman, Tidak ada celah yang memungkinkan terjadinya ancaman dan akses illegal yang berpotensi merusak sistem. Diantaranya adalah sebagai berikut :

Tabel 2. Daftar Peringatan Kesalahan

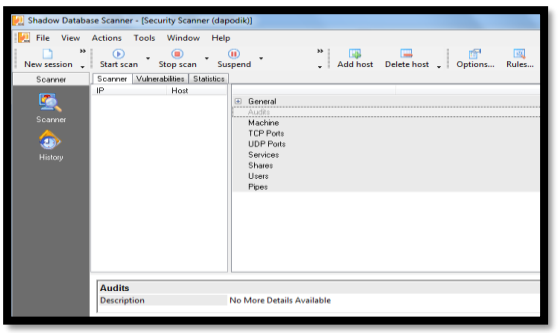
No	Pesan (*Alert Found*)	Peringatan (*Variations*)
1	*Apache Chunked-Encoding Memory Corruption Vulnerability:*	0
2	*PHP HTML Entity Encoder Heap Overflow Vulnerability*	0
3	*PHP multiple vulnerabilities*	0
4	*PHP POST file upload buffer overflow vulnerabilities*	0
5	*PHP unspecified remote arbitrary file upload vulnerability*	0
6	*PHP Zend_Hash_Del_Key_Or_Index vulnerability*	0
7	*SQL Injection*	0
8	*Unfiltered Header Injection in Apache 1.3.34/2.0.57/2.2.1*	0
9	*Apache Error Log Escape Sequence Injection Vulnerability*	0
10	*Apache version 6.0*	0
11	*PHP mail function ASCII control character header spoofingvulnerability*	0
12	*PHP socket_iovec_alloc() integer overflow*	0
13	*PHP4 multiple vulnerabilities*	0
14	*Apache version up to 1.3.33 htpasswd local overflow*	0
15	*TRACE Method Enabled*	0
16	*Password type input with autocomplete enabled*	0
	Total Alert Found	0

Hasil pengujian Peringatan keamanan terhadap web server Dapodik adalah 0 Aman, Tidak ada celah satu pun yang memungkinkan terjadinya akses illegal yang bisa merusak sistem.

Analisis keamanan pada sisi *database server* dalam penelitian ini dilakukan dengan menggunakan *software Shadow Database Scanner* versi 7.75. Aspek-aspek yang dianalisis meliputi :

a. *IP address*
b. *Host name*
c. *Average ping response*
d. *TCP port*

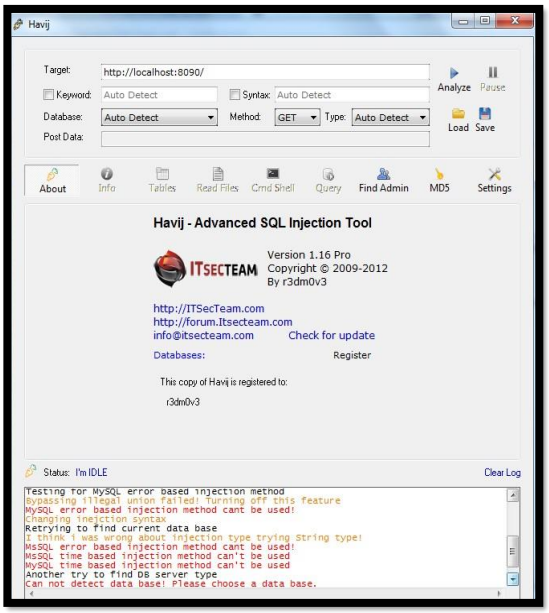
Hasil analisis menggunakan *Shadows Database Scanner* adalah sebagai berikut :



Gambar 8. Hasil Analisis SDS *Database Server*

Dari hasil pengujian menunjukan tidak ada database pada aplikasi dapodik, karena memang benar untuk database aplikasi ini berada di server pusat tidak berada di aplikasi, ini menunjukan bahwa database server dapodik memang tidak terdeteksi oleh tools tersebut.

Hasil analisis menggunakan *havij* sebagai berikut :



Gambar 9. Hasil Analisis *Havij Database Server*

Berdasarkan pengujian menggunakan software *Havij*, *database* aplikasi dapodik tidak dapat di injeksi menggunakan Havij karena aplikasi web dapodik menggunakan method POST yang berfungsi untuk menyembunyikan variable dan value, halaman dapodik yang di injeksi tidak memiliki keterkitan dengan basis data dan halaman dapodik tersebut juga bersifat statis.

Bersasarkan dari hasil Pengujian Keseluruhan Tingkat Keamanan yang dilakukan pada Aplikasi Dapodik menggunakan *Software Tools Acunnetix Vulnerability Scanner, Shadows Database Scanner*, dan *Havij* tidak ada serangan atau celah satupun yang memungkinkan terjadi akses illegal yang bertpotensi merusak sistem, Acunnetix Vulnerability Scanner tidak menemukan celah atau

akses illegal yang berpotensi merusak sistem yang dilakukan menggunakan SQL Injection pada aplikasi tersebut. *Shadows Database Scanner* tidak mendeteksi / menemukan basis data *database* karena aplikasi dapodik bersifat sinkronisasi data artinya mengirimkan basis data atau data utama dapodik, sinkronisasi juga bisa di sebut sebagai penyatuan database sehingga database berada di server pusat tidak pada aplikasi. Kemudian *database* aplikasi dapodik tidak dapat di injeksi menggunakan *Havij* karena aplikasi web dapodik menggunakan method POST yang berfungsi untuk menyembunyikan variable dan value, halaman dapodik yang di injeksi tidak memiliki keterkitan dengan basis data dan halaman dapodik. Maka tingkat Keamanan aplikasi dan *database server* pada Aplikasi Dapodik menunjukan Aman.

3. KESIMPULAN

Penelitian analisis *Web Vulnerabilities* untuk meningkatkan keamanan aplikasi Dapodik dapat disimpulkan sebagai berikut :

1. Penelitian ini berhasil melakukan analisis terhadap aspek-aspek keamanan system yang meliputi keamanan web server, program aplikasi dan pada database yang digunkan pada system aplikasi Dapodik.
2. Penyerangan dengan Teknik *SQL Injection* tidak bisa dilakukan pada aplikasi Dapodik.
3. Tingkat ancaman terhadap aplikasi Dapodik berada pada level 0 (aman). Aplikasi Dapodik belum ditemukan celah yang memungkinkan terjadinya ancaman dan akses illegal yang berpotensi merusak sistem. Pengujian menggunakan *Acunnetix Vulnerability Scanner*, *Shadows Datavase Scanner*, dan *Havij*.
4. Usulan kebijakan keamanan aplikasi Dapodik bersifat dinamis dan menyesuikan dengan perkembangan teknologi dan informasi.

PUSTAKA

Kesharwani, Pawan, Sudhanshu Shekhar Pandey, Vishal Dixit, Lokendra Kumaer Tiwari. 2018. “A Study on Penetration Testing Using Metasploit Framework”. International Research Journal of Engineering and Technology (IRJET) Volume 05 Issue 12.

Sutanta, Edhy. 2012. Analisis Keamanan Sistem Aplikasi. AKPRIND Yogyakarta.

Shar, L.K. and H. B.K. Tan. 2013.”Defeating SQL Injection”, Computer (Long. Beach. Calif)., Vol. 46 No.3 pp 66-77.

Singh, P., K. Thevar, P. Shetty, and B. Shaikh. 2015. “Detection of SQL Injection and XSS Vulnerability in Web Application”, No.3 pp 16-21.

Wahyudi. 2019. “Analisa Pengujian Kerentanan Terhadap Web Server SIMAK (Studi Kasus STMIK Kharisma Karawang). Jurnal Teknologi Informasi Vol. 5 No. 1 Juni.

Yudhistira, Alifiandy. 2012. Analisis KEamanan. Universitas Indonesia.