

**UNIVERSITAS INDONESIA**

**PERANCANGAN PROSEDUR KEAMANAN PADA PENGEMBANGAN APLIKASI WEB DI BADAN PUSAT STATISTIK**

**TUGAS MPPI**

**NIA DWI RAHAYUNINGTYAS**

**1806255903**

**FAKULTAS ILMU KOMPUTER**

**PROGRAM STUDI MAGISTER TEKNOLOGI INFORMASI**

**JAKARTA**

**2019**

**ABSTRAK**

Nama : Nia Dwi Rahayuningtyas

Program Studi : Magister Teknologi Informasi

Judul : Perancangan Prosedur Kemanan pada Pengembangan Aplikasi Web di Badan Pusat Statistik

Pembimbing : -

BPS (Badan Pusat Statistik) sebagai penyedia data turut menyelenggaraan e-government yang menggunakan teknologi internet dalam memberikan pelayanan. Penggunaan internet memiliki dampak positif dan dampak negatif. Ancaman kejahatan siber dapat dilakukan dengan cara memanfaatkan vulnerability atau celah keamanan yang terdapat pada aplikasi web e-government. Statistik vulnerability aplikasi website dengan domain .bps.go.id pada tahun 2018 menunjukkan bahwa masih banyak terdapat vulnerability sebesar 53% dengan tingkat vulnerability tinggi. Tercatat pada tahun 2017 serangan web defacement berhasil mengubah konten pada 55 subdomain .bps.go.id. Sedangkan pada akhir 2018, serangan dengan jenis yang sama berhasil mengubah konten pada 99 website operasional BPS dan website BPS Provinsi dan Kabupaten/Kota. Pengubahan konten tersebut menyebabkan Data (publikasi) tidak dapat diakses atau data yang diakses bukanlah data yang sebenarnya, *Down time* mengakibatkan layanan terhenti (*non-availability*), sistem elektronik layanan publik gagal beroperasi, penurunan kinerja operasional di BPS dan kepercayaan publik terhadap layanan di BPS. Akar permasalahan dari insiden tersebut adalah adanya vulnerability tinggi pada aplikasi web akibat belum ada prosedur keamanan pada pengembangan aplikasi web. Penelitian ini bertujuan untuk menghasilkan rancangan prosedur keamanan pada pengembangan aplikasi web di BPS. Penelitian ini menggunakan *Mixed-method* dengan teknik observasi dan studi dokumen untuk pengumpulan data. Studi literatur dilakukan pada sejumlah dokumen terkait kebijakan dan standar keamanan pengembangan aplikasi web. Penelitian ini akan menghasilkan rancangan prosedur keamanan aplikasi web yang secara umum dapat diimplementasikan pada pengembangan aplikasi web terutama web e-government. Penetian ini akan memberikan manfaat bagi developer, organisasi, dan berkontribusi bagi penelitian terkait keamanan pengembangan aplikasi web.

Kata kunci: prosedur keamanan, pengembangan aplikasi web, vulnerability, OWASP, ISO/IEC 27001

# **DAFTAR ISI**

[**DAFTAR ISI** iii](#_Toc11665326)

[**DAFTAR GAMBAR** iv](#_Toc11665327)

[**DAFTAR TABEL** v](#_Toc11665328)

[**DAFTAR LAMPIRAN** vi](#_Toc11665329)

[**BAB 1** 1](#_Toc11665330)

[**1.1** **Latar Belakang Penelitian** 1](#_Toc11665331)

[**1.2** **Perumusan Masalah** 5](#_Toc11665332)

[**1.3** **Ruang Lingkup Masalah** 10](#_Toc11665333)

[**1.4** **Tujuan Penelitian** 10](#_Toc11665334)

[**1.5** **Manfaat Penelitian** 10](#_Toc11665335)

[**1.6** **Sistematika Penulisan** 10](#_Toc11665336)

[**BAB 2 TINJAUAN PUSTAKA** 12](#_Toc11665337)

[**2.1.** **Landasan Teori** 13](#_Toc11665338)

[**2.1.1.** **Definisi Kebijakan, Standar, Pedoman, dan Prosedur** 13](#_Toc11665339)

[**2.1.2.** **Aplikasi Web** 14](#_Toc11665340)

[**2.1.3.** **Metodologi Pengembangan Aplikasi Web** 15](#_Toc11665341)

[**2.1.4.** **Keamanan Informasi** 16](#_Toc11665342)

[**2.1.5.** **Keamanan Aplikasi Web** 16](#_Toc11665343)

[**2.1.6.** **Ancaman Keamanan Aplikasi Web** 17](#_Toc11665344)

[**2.1.7.** ***Vulnerability* Aplikasi Web** 18](#_Toc11665345)

[**2.1.8.** **ISO/IEC 27001** 19](#_Toc11665346)

[**2.1.9.** **OWASP (*Open Web Application Security Project*)** 19](#_Toc11665347)

[**2.2.** **Penelitian Sebelumnya** 26](#_Toc11665348)

[**2.3.** **Metodologi** 33](#_Toc11665349)

[**2.4.** ***Theoretical Framework*** 37](#_Toc11665350)

[**BAB 3 METODOLOGI PENELITIAN** 39](#_Toc11665351)

[**3.1.** **Desain Penelitian** 39](#_Toc11665352)

[**3.2.** **Alur Penelitian** 40](#_Toc11665353)

[**3.3.** **Pengumpulan Data** 42](#_Toc11665354)

[**3.4.** **Penyusunan Rekomendasi** 42](#_Toc11665355)

[**3.5.** **Jadwal Penelitian** 43](#_Toc11665356)

[**DAFTAR PUSTAKA** 44](#_Toc11665357)

[**LAMPIRAN** 47](#_Toc11665358)

# **DAFTAR GAMBAR**

[Gambar 1.1 Statistik Insiden .go.id Tahun 2017 2](#_Toc11663668)

[Gambar 1.2 Statistik Vulnerability pada Aplikasi Web .bps.go.id Tahun 2018 3](#_Toc11663669)

[Gambar 1.3 Web Defacement pada Web BPS 4](#_Toc11663670)

[Gambar 1.4 Analisis Kesenjangan 6](#_Toc11663671)

[Gambar 1.5 Diagram Mind Map 8](#_Toc11663672)

[Gambar 2.1 Diagram Proses Seleksi Menggunakan PRISMA 12](#_Toc11663673)

[Gambar 2.2 Keterkaitan antara Kebijakan, Standar, Prosedur, dan Pedoman 14](#_Toc11663674)

[Gambar 2.3 Arsitektur Aplikasi Web 15](#_Toc11663675)

[Gambar 2.4 Theoretical Framework 37](#_Toc11663676)

[Gambar 3.1 Tahapan Penelitian 42](#_Toc11663677)

# **DAFTAR TABEL**

[Tabel 2.1 Perbandingan dengan Penelitian-penelitian Sebelumnya 28](#_Toc11663678)

[Tabel 2.2 Perbandingan Framework dan Metodologi Olah Data 33](#_Toc11663679)

[Tabel 3.1 Desain Penelitian 40](#_Toc11663680)

# **DAFTAR LAMPIRAN**

[**Lampiran 1** Transkrip Wawancara dengan Kepala Subdirektorat Jaringan Komunikasi Data 47](#_Toc11663681)

[**Lampiran 2** Transkrip Wawancara dengan Staf Jaringan Komunikasi Data 49](#_Toc11663682)

[**Lampiran 3** Hasil Self Assessment BPS Menggunakan Index KAMI 3.1 Tahun 2016 51](#_Toc11663683)

[**Lampiran 4** Raw Log Insiden Intrusi 99 Website Oktober 2018 52](#_Toc11663684)

[**Lampiran 5** Laporan Kronologi Insiden Intrusi Website Oktober 2018 54](#_Toc11663685)

[**Lampiran 6** Laporan Kronologi Insiden Intrusi Website Januari 2017 55](#_Toc11663686)

[**Lampiran 7** Laporan Hasil Vulnerability Assessment Web .bps.go.id Desember 2018 56](#_Toc11663687)

[**Lampiran 8** Transkrip Wawancara dengan Kepala Seksi Pengembangan Jaringan Komunikasi Data 57](#_Toc11663688)

[**Lampiran 9** Pernyataan Kategori Sistem Elektronik (Snapshoot www.bps.go.id) 59](#_Toc11663689)

[**Lampiran 10** Visi-Misi dan Sasaran TI BPS (Perka BPS No 94 Tahun 2018) 60](#_Toc11663690)

**Lampiran 11** Contoh Lembar Observasi Developer ........................................... 63

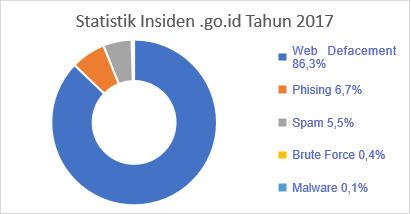
**PENDAHULUAN**

Bab ini berisi latar belakang penelitian, perumusan masalah, ruang lingkup masalah, tujuan serta manfaat penelitian dan sistematika penulisan penelitian.

## **Latar Belakang Penelitian**

Keterbukaan informasi publik menjadi tuntutan bagi pemerintah dalam penerapan *e-government*. Penyelenggaraan pelayanan publik dengan memanfaatkan teknologi informasi didasarkan pada Undang-undang Nomor 14 Tahun 2008 tentang Keterbukaan Informasi Publik. Teknologi informasi telah memberikan kontribusi yang besar terhadap bertambahnya arus informasi dan transaksi online di dunia maya. Pemerintah telah mengatur penyelenggaraan sistem dan transaksi elektronik dalam UU-ITE (UU Nomor 11 Tahun 2008) dan Peraturan Pemerintah Republik Indonesia Nomor 82 Tahun 2012.

Dalam sebuah pemerintahan, *e-government* menjadi sangat penting untuk menjaga interaksi antara pemerintah dan masyarakat dalam hal pelayanan (Nwaeze, 2017). *E-government* ditujukan untuk meningkatkan interaksi antara pemerintah dan masyarakat (G2C) dengan cara yang lebih ramah, lebih nyaman, transparan, dan hemat biaya (Nwaeze, 2017). Penyelenggaraan *e-government* yang menggunakan teknologi internet memiliki dampak positif dan dampak negatif. Teknologi internet dalam bentuk aplikasi website dapat memudahkan publik dalam mendapatkan pelayanan. Akan tetapi, menurut laman <https://bssn.go.id> penggunaan internet juga diikuti oleh dampak negatif berupa ancaman kejahatan siber (*cyber threats*). Ancaman kejahatan siber ini dilakukan dengan cara memanfaatkan kerentanan atau celah keamanan yang terdapat pada sistem *e-government*. Ancaman kejahatan ini jika dilakukan akan menyebabkan terjadinya insiden pelanggaran keamanan (*security breaches*). Fasilitas Pemerintahan termasuk dalam sektor strategis di Indonesia, Amerika, dan termasuk critical infrastructure di Kanada. Resiko yang bisa timbul adalah kegagalan beroperasinya sistem elektronik layanan publik pemerintahan & layanan administrasi, kebocoran, pencurian dan perubahan informasi dan data pribadi dan lain-lain (Nwaeze, 2017).



Gambar 1.1 Statistik Insiden .go.id Tahun 2017

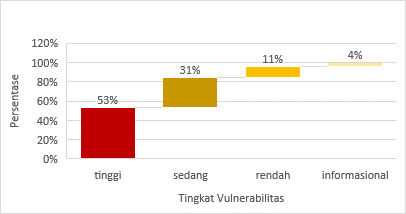
Sumber: (ID-GOVCSIRT, 2017)

Berdasarkan statistik insiden yang dirilis oleh *Indonesia Government Security Incident Response Team* (ID-GOVCSIRT, 2017) beberapa serangan terhadap website .go.id yang dilaporkan paling banyak adalah *web defacement* dengan persentase 86,3 %. Selain itu berupa *phising, spam, brute force*, dan *mallware* dengan persentase masing-masing seperti pada Gambar 1.1 *Statistik Insiden .go.id Tahun 2017*.

Nwaeze (2017) menyebutkan dalam sistem *e-Government*, masalah privasi informasi menjadi pusat perhatian ketika sistem mengumpulkan, menyimpan, dan memproses data individu untuk menyediakan berbagai layanan online. Pada Agustus 2015 terjadi kasus kebocoran data pada Layanan Pendapatan Internal Pemerintah (IRS) A.S. yang mengakibatkan kerugian sebesar $39 juta. Hal ini dapat digunakan sebagai contoh yang menggambarkan pentingnya keamanan dan privasi dalam sistem *e-Government*.

BPS sebagai instansi penyedia data telah memberikan layanan publik berupa pelayanan data melalui aplikasi berbasis internet, yaitu website dan aplikasi *mobile*. Beberapa website di BPS baik dengan fungsi pelayanan, operasional, maupun strategis tidak luput dari serangan tersebut. Hampir setiap tahun website-website tersebut mengalami serangan berupa *web defacement*. *Web defacement* merupakan suatu upaya yang dilakukan secara tidak sah untuk mengganti visual atau tampilan halaman utama sebuah web (https://bssn.go.id).

Tercatat pada tahun 2017 serangan *web defacement* berhasil mengubah konten pada 55 subdomain .bps.go.id. Sedangkan pada akhir 2018, serangan dengan jenis yang sama berhasil mengubah konten pada 99 website operasional BPS dan website BPS Provinsi dan Kabupaten/Kota. Menurut laporan kronologi (Lampiran 5 dan Lampiran 6) serangan tersebut dilakukan dengan menyisipkan sebuah file berisi *script* berbahaya yang mengubah konten website. Hasil investigasi menyatakan bahwa penyebab utama terjadinya insiden adalah terdapat banyak vulnerability (celah keamanan) pada aplikasi web.

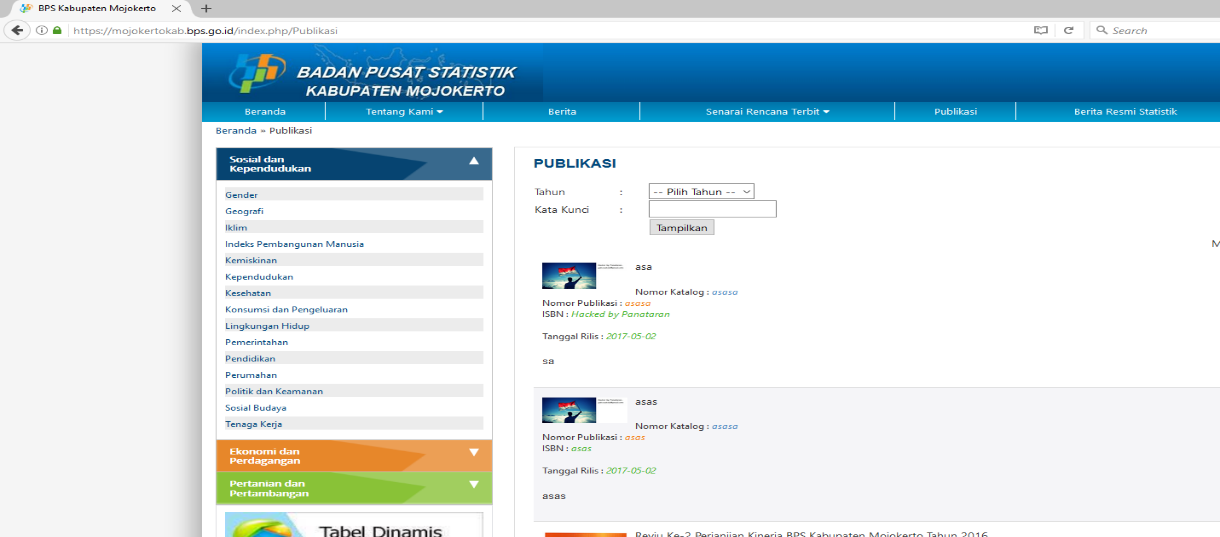


Gambar 1.2 Statistik Vulnerability pada Aplikasi Web .bps.go.id Tahun 2018

(Sumber: Laporan Hasil Vulnerability Assessment Web .bps.go.id Desember 2018, diolah kembali (Lampiran 7))

Statistik vulnerability aplikasi website dengan domain .bps.go.id pada tahun 2018 menunjukkan bahwa masih banyak terdapat vulnerability sebesar 53% dengan tingkat vulnerability tinggi. Berdarkan hasil scan aplikasi web menggunakan Acunetix didapatkan lima besar vulnerability kritikal antara lain *Blind SQL Injection, Cross site scripting, SQL injection, SSL 2.0 deprecated protocol,* dan *The DROWN attack (SSLv2 supported)*.

Vulnerability menjadi isu strategis di BPS karena, menurut Nguyen-tuong (2015) dan Van Goethem (2014) adanya vulnerability ini dapat menyebabkan beberapa resiko krusial terjadi pada instansi. Adanya resiko terhadap pencurian data (data terkait privasi dan strategis), pengubahan konten, dan ketiadaan layanan (*non-availability*) menjadi perhatian utama karena BPS merupakan instansi penyedia data bagi pemerintah untuk dijadikan acuan dalam penentuan kebijakan. Data di BPS termasuk dalam kategori data pribadi yang bersifat individu dan/atau data pribadi yang terkait dengan kepemilikan badan usaha yang sifatnya rahasia dan terbatas (Lampiran 9). Dalam Alazman, et. al. (2018) disebutkan bahwa sangat penting bagi instansi atau perusahaan dalam melindungi datanya agar terjaga *confidentiality, integrity, dan availability* (CIA). Gambar 1.3 Web Defacement *pada Web BPS* adalah salah satu contoh pengubahan konten yang terjadi pada insiden keamanan di BPS. Pengubahan konten yang terjadi menyebabkan beberapa data (publikasi) tidak dapat diakses atau data yang diakses bukanlah data yang sebenarnya.



Gambar 1.3 Web Defacement pada Web BPS

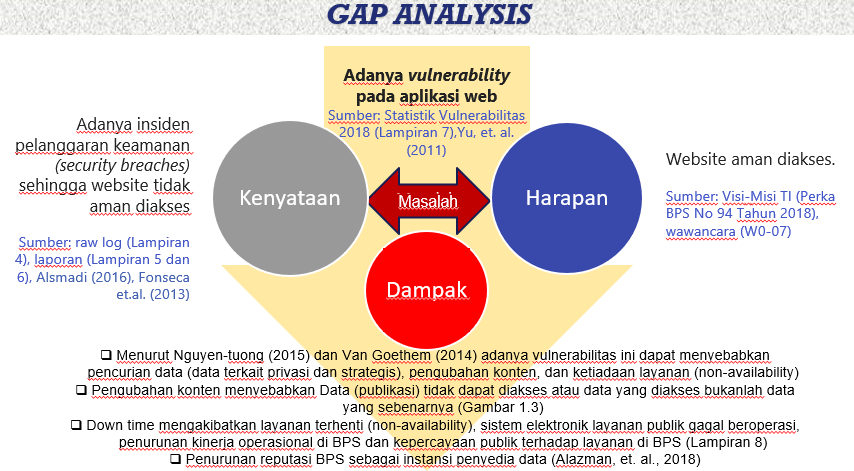
Pada laporan kronologi (Lampiran 5 dan Lampiran 6) disebutkan bahwa upaya penanganan dilakukan dengan menginvestigasi log sistem dan mematikan layanan dalam beberapa waktu (*down time*). *Down time* ini mengakibatkan layanan terhenti (*non-availability*) dan tentu saja hal ini selain berdampak signifikan terhadap penurunan kinerja operasional di BPS, juga menurunkan kepercayaan publik terhadap layanan di BPS dan pada akhirnya berdampak terhadap reputasi BPS sebagai instansi penyedia data (Lampiran 8). Jika instansi/perusahaan gagal melindungi aspek CIA dapat mengakibatkan kerugian, permasalahan hukum dan dapat merusak reputasi instansi atau perusahaan (Alazman, et. al., 2018).

Hasil *self assessment* yang dilakukan BPS menggunakan index KAMI Tahun 2016 menyatakan bahwa indeks keamanan informasi mendapatkan nilai 249 dengan kategori “Tidak Layak” (Lampiran 3). Beberapa komponen yang memiliki nilai indeks sangat minim adalah kategori pengelolaan risiko, tata kelola, dan kerangka kerja keamanan informasi. Pengelolaan aset informasi dan pengelolaan risiko secara umum masih dalam tahap perencanaan. Aset informasi, seperti aplikasi atau sistem informasi, harus dilakukan pengelolaan dengan baik.

Undang-undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (UU ITE) menyebutkan “Setiap penyelenggara sistem elektronik harus menyelenggarakan sistem elektronik secara andal dan aman serta bertanggung jawab terhadap beroperasinya sistem elektronik sebagaimana mestinya” (Pasal 15 ayat 1). Peraturan Pemerintah Nomor 82 Tahun 2012 tentang Penyelenggaraan Sistem dan Transaksi Elektronik menyebutkan “Penyelenggara sistem elektronik wajib melakukan pengamanan terhadap komponen sistem elektronik” (pasal 19). BPS sebagai instansi pemerintah penyelenggara sistem elektronik yang memberikan pelayanan data, memiliki Visi-Misi TI yaitu "Pengguna mendapatkan manfaat dari layanan TI BPS yang prima dan inovatif secara aman, efektif, efisien, terpercaya, dan handal" (Perka BPS No 94 Tahun 2018). Pasal 7 (Perka BPS No 73 tahun 2016) mengenai prinsip keamanan TI menyatakan bahwa, (1) keamanan TI merupakan kewajiban semua pihak, (2) adopsi keamanan TI dengan pendekatan berbasis risiko, (3) fokus pada aplikasi yang penting. Adopsi keamanan TI dengan pendekatan berbasis risiko maksudnya adalah pengelolaan ancaman terhadap keamanan TI harus mengadopsi proses manajemen risiko untuk dapat mengelola dan memprioritaskan penanganan risiko. Berdasarkan ketentuan tersebut, pengelolaan ancaman terhadap keamanan TI dibutuhkan sehingga risiko keamanan TI dapat diidentifikasi, dianalisa, dimitigasi, dan ditangani secara efektif.

## **Perumusan Masalah**

Analisis kesenjangan pada Gambar 1.4 *Analisis Kesenjangan*dilakukan untuk mengidentifikasi masalah dan dampaknya. BPS mempunyai harapan bahwa website yang diakses oleh publik/pengguna adalah website yang aman. Website yang aman adalah website yang bebas kerentanan atau celah keamanan (*vulnerability*) (Fonseca et. al., 2013) dan bebas dari *script* yang berbahaya (Alsmadi, 2018). *Vulnerability* pada aplikasi web dapat memberikan peluang terhadap serangan siber untuk mencuri, memodifikasi, dan merusak data (Yu, et. al., 2011). Gartner (Yu, et. al., 2011) menyebutkan bahwa aplikasi web yang memiliki masalah keamanan dapat menjadi masalah yang membahayakan yang berimbas pada perasaan aman pengguna dalam mengakses web.

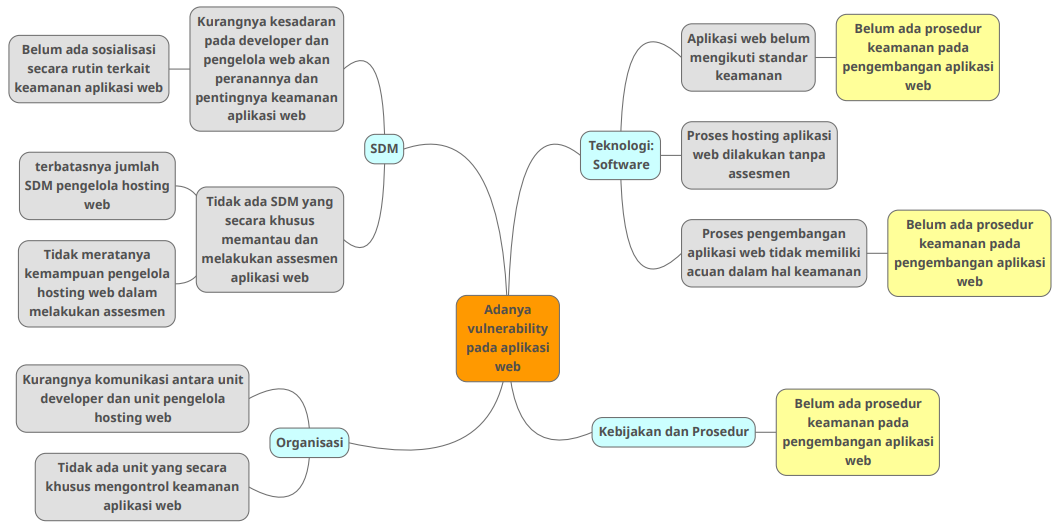


Gambar 1.4 Analisis Kesenjangan

Akan tetapi, pada kenyataannya website BPS masih dalam kategori tidak aman dibuktikan dengan adanya insiden pelanggaran keamanan (*security breach*) yang terjadi beberapa kali. Pelanggaran keamanan terjadi karena adanya aplikasi website di BPS masih memiliki banyak *vulnerability* seperti yang dijalaskan pada Gambar 1.2 *Statistik Vulnerability pada Aplikasi Web .bps.go.id Tahun 2018*.

Insiden pelanggaran keamanan dapat menyebabkan terhentinya kegiatan operasional akibat website yang tidak dapat diakses (W3-05). Terhentinya kegiatan operasional ini dapat berdampak pada terhambatnya pelayanan publik, terutama pelayanan yang menggunakan website, sehingga kepercayaan publik menurun. Insiden yang terjadi pada Bulan Oktober 2018 mengakibatkan perubahan konten website, salah satunya website pelayanan data silastik.bps.go.id (Lampiran 4 dan Lampiran 5). Website ini merupakan website utama BPS yang digunakan dalam kegiatan operasional melayani permintaan data dan konsultasi. Akibat terjadinya serangan ini, pihak pengelola melakukan maintenance dengan mematikan layanan dalam beberapa waktu, sehingga pelayanan pun menjadi terhambat. Kenyataan ini pada akhirnya berdampak buruk terhadap reputasi BPS sebagai penyedia layanan publik (Alsmadi, 2016). Oleh karena itu diperlukan adanya mekanisme yang tepat untuk mengatasi masalah ini.

Berdasarkan kondisi dan permasalahan serta urgensinya yang diuraikan di atas, selanjutnya dilakukan analisis akar permasalahan menggunakan diagram *mind map*. Dalam buku *Principle of information security* (Whitman, Michael, M.J. Herbert, 2012), domain untuk melihat akar permasalahan pada keamanan sebuah sistem informasi terdiri dari teknologi (perangkat keras dan lunak), prosedur, manusia, dan data.Sedangkan, menurut Achmadi (2018) keamanan system informasi dipengaruhi oleh faktor sumber daya manusia (SDM), teknologi, kebijakan dan prosedur, serta organisasi. Pada penelitian ini domain yang digunakan adalah empat domain utama yang didapatkan dari teori Whitman dengan penambahan domain organisasi. Berikut adalah analisis dari permasalahan dengan menggunakan diagram *mind map.*



Gambar 1.5 Diagram Mind Map

1. Domain SDM (*people*)

Terdapatnya vulnerability pada aplikasi web disebabkan karena kurangnya kesadaran dari developer dan pengelola web akan pentingnya peranan mereka dalam keamanan aplikasi web (W0-05). Hal ini disebabkan karena belum adanya sosialisasi atau pelatihan terkait keamanan web. Selain itu disebabkan juga oleh ketiadaan SDM yang berperan dalam mengassess aplikasi karena keterbatasan pengelola layanan hosting (Lakin Direktorat SIS, 2018).

1. Domain Teknologi: *Software*

Domain Teknologi mencakup aspek software dan hardware. Aspek software merupakan aplikasi web yang dibangun, sedangkan akspek hardware tidak dibahas pada penelitian ini. Dari aspek tersebut dilakukan wawancara dan observasi dokumen sehingga menghasilkan akar permasalahan yaitu banyaknya *vulnerability* (kerentanan) pada aplikasi website yang dibangun (W1-01). Aplikasi website dibangun oleh *developer* yang tersebar di unit-unit kerja BPS, sehingga pengembangan aplikasi cenderung hanya memenuhi persyaratan fungsional tanpa memperhatikan persyaratan non-fungsional, salah satunya terkait keamanan aplikasi. Aplikasi website yang dihosting pada server belum memenuhi kriteria keamanan karena belum adanya prosedur keamanan pada pengembangan aplikasi web (W0-06).

1. Domain Kebijakan dan Prosedur

Hal ini disebabkan karena belum adanya prosedur keamanan pada pengembangan aplikasi web (W0-06).

1. Domain Organisasi

Kurangnya komunikasi antara *developer* dan tim pengelola *hosting* web (W1-03) menjadi penyebab aplikasi web yang dibangun hanya didasarkan pada funfsionalitas tanpa memperhatikan aspek keamanan.

Setelah dilakukan analisis lebih lanjut terhadap akar permasalahan, observasi dokumen, dan wawancara, maka penelitian ini berfokus pada akar permasalahan mendasar yaitu “**Belum ada prosedur keamanan pada pengembangan aplikasi web**”. Akar permasalahan tersebut dipilih agar selaras dengan sasaran TI poin G2 sebagai turunan visi-misi TI BPS yang ditunagkan dalam Perka BPS No 94 Tahun 2018 yaitu "menyediakan pengumpulan, penyimpanan, dan diseminasi data yang aman", Poin 4.6.1 tentang kebijakan dan standar area keamanan dan poin 4.6.10 tentang area keamanan aplikasi (Lampiran 10 *Visi-Misi dan Sasaran TI BPS (Perka BPS No 94 Tahun 2018)*. Berdasarkan akar permasalahan yang dipilih sebagai forkus utama penelitian, maka pertanyaan penelitian yang diajukan adalah **“Bagaimana rancangan prosedur keamanan yang dapat diterapkan dalam pengembangan aplikasi web di BPS?”.**

## **Ruang Lingkup Masalah**

Ruang lingkup masalah pada penelitian ini adalah standar keamanan yang diterapkan pada setiap fase pengembangan aplikasi website di BPS.

## **Tujuan Penelitian**

Tujuan penelitian ini adalah untuk menghasilkan rancangan prosedur keamanan pada pengembangan aplikasi web di BPS.

## **Manfaat Penelitian**

Hasil penelitian dapat memberikan manfaat sebagai berikut:

1. Bagi *developer*, dapat menjadi acuan dalam pengembangan aplikasi web yang aman.
2. Bagi organisasai, dapat meningkatkan kualitas aplikasi web sehingga kepercayaan masyarakat terhadap layanan BPS meningkat, dan pada akhirnya meningkatkan reputasi BPS di mata publik.
3. Bagi kalangan akademisi, dapat dijadikan sebagai acuan dan masukan untuk peningkatan kualitas keamanan aplikasi web.

## **Sistematika Penulisan**

Sistematika dalam penulisan karya akhir ini disusun sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang penelitian, perumusan masalah, ruang lingkup maslah, tujuan, manfaat, dan sistematika penulisan

BAB 2 LANDASAN TEORI

Bab ini berisi tentang landasan teori, metode, model, best practice, serta pertaturan yang terkait dengan penelitian ini. Selain itu, bab ini juga membahas mengenai penelitian sebelumnya yang relevan dengan penelitian ini

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi metodologi yang digunakan dalam penelitian mulai dari masukan, proses sampai dengan keluaran yang diharapkan.

BAB 4 PROFIL ORGANISASI

Bab ini berisi profil organisasi tempat dilakukannya penelitian yaitu Subdit Integrasi Pengolahan Data BPS.

BAB 5 ANALISIS

Bab ini berisi hasil analisis kondisi saat ini serta rekomenadasi perbaikan proses pengujian pada pengembangan perangkat lunak.

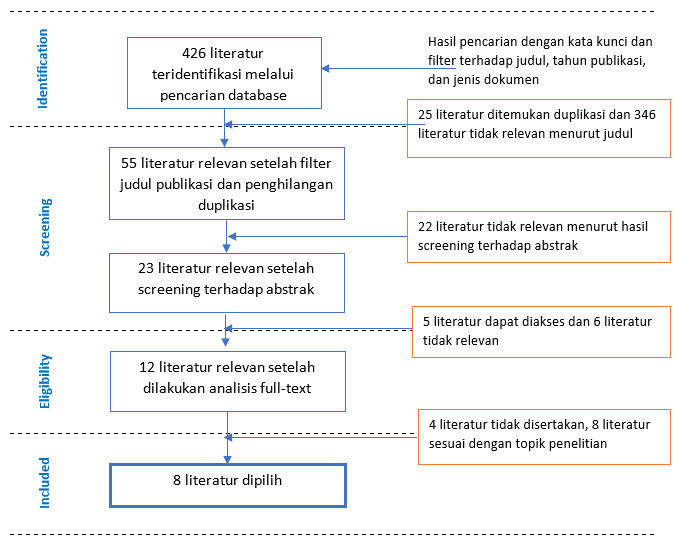
BAB 6 PENUTUP

Bab ini berisi saran dan kesimpulan dari penelitian yang dilakukan serta penelitian lanjutan yang dapat dilakukan.

# **TINJAUAN PUSTAKA**

Bab ini berisi mengenai landasan teori terkait penelitian ini, penelitian-penelitian sebelumnya dan *theoretical framework*. Metode yang digunakan oleh penulis dalam melakukan tinjauan pustaka adalah menggunakan PRISMA (*Preffered Reporting Items for Systematic Reviews and Meta-Analyses*). PRISMA adalah protokol tinjauan sistematis dan metaanalisis (Moher, et. al., 2009) yang terdiri dari 27 item daftar periksa termasuk judul, pengantar, metode, dan lain-lain.

Pencarian literatur dilakukan pada *Web of Knowledge Database*, ACM, IEEE, Scopus, Google Scholar, dan Science Direct. Strategi pencarian yang dilakukan dalam pencarian literatur pada beberapa basis data adalah menggunakan kata kunci **“(policy OR standard OR procedure) AND (web application OR web application development) AND security OR (owasp in government)”**. Seleksi literatur dilakukan dengan filter terhadap judul dan abstrak berdasarkan kata kunci, tahun antara 2015 sampai 2019, dan jenis artikel berupa jurnal atau conference atau buku berbahasa inggris. Kemudian dilakukan seleksi dengan analisis isi literatur. Berikut gambar diagram proses seleksi menggunakan PRISMA:



Gambar 2.1 Diagram Proses Seleksi Menggunakan PRISMA

## **Landasan Teori**

Landasan teori berisi definisi mengenai kebijakan, standar, pedoman, dan prosedur, aplikasi web, pengembangan aplikasi web, ancaman keamanan aplikasi web, *vulnerability* aplikasi web, standar keamanan aplikasi web, dan OWASP.

### **Definisi Kebijakan, Standar, Pedoman, dan Prosedur**

Menurut Cannon (2011), kebijakan merupakan dokumen tingkat tinggi yang memiliki kekuasaan untuk memaksa kerjasama. Kebijakan adalah dokumen sederhana yang menyatakan tujuan kontrol tingkat tinggi tertentu untuk kesuksesan organisasi. Kedudukan orang yang menetapkan kebijakan akan menentukan lingkup penerapan kebijakan.

Standar merupakan dokumen level menengah berisi kontrol pengukuran untuk memastikan penerapan yang seragam untuk mendukung kebijakan. Standar mengidentifikasi kontrol tertentu yang diperlukan untuk kepatuhan. Namun, standar tidak berisi alur kerja untuk kepatuhan.

Pedoman dimaksudkan untuk memberikan saran mengenai bagaimana mencapai tujuan organisasi bila tidak terdapat atau kurangnya standar. Tujuan pedoman adalah untuk memberikan informasi yang akan membantu pengambilan keputusan mengenai tujuan yang diinginkan, alternatif yang menguntungkan, dan tindakan yang tidak diperlukan. Pedoman tidak harus diikuti karena petunjuk yang disediakan biasanya belum lengkap. Pengguna harus mengadaptasi atau membuang bagian yang tidak diperlukan agar sesuai dengan kebutuhan.

Prosedur menyediakan alur kerja atau pekerjaan spesifik yang perlu dilakukan untuk memperoleh kepatuhan minimal terhadap standar. Setiap langkah detail dituliskan dari awal sampai akhir. Prosedur yang baik termasuk langkah umum penyelesaian masalah bila terdapat masalah umum yang dihadapi pengguna. Praktek terbaik dapat membantu pengguna membuat prosedur yang diinginkan. Tujuan dari prosedur adalah untuk menjaga kontrol tertinggi yang mungkin atas hasil. Kepatuhan terhadap prosedur yang dibuat wajib dilakukan untuk memastikan konsistensi dan prosedur.

Keterkaitan antara kebijakan, standar, prosedur dan pedoman dapat dilihat pada Gambar 2.2.



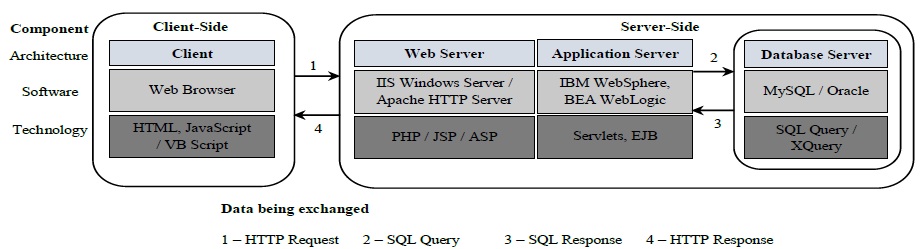
Gambar 2.2 Keterkaitan antara Kebijakan, Standar, Prosedur, dan Pedoman

(Sumber: Cannon, 2011)

### **Aplikasi Web**

Aplikasi web adalah sebuah sistem informasi *Platform World Wide Web* (WWW) yang di dalamnya terdapat prinsip-prinsip komputer yang disebut sebagai *client-server* (Molina, et. al., 2016). Aplikasi web terdiri dari proses pengguna melakukan permintaan ke sebuah program pada server dan server bertanggung jawab untuk memberikan respon terhadap permintaan tersebut. Aplikasi web menghasilkan serangkaian halaman secara dinamis dalam format standar, seperti HTML (*HyperText Markup Language*), atau XHTML (*Extensible HyperText Markup Language*), yang didukung oleh browser web. Hal ini diperlukan dalam penggunaan bahasa pemrograman yang diimplementasikan dalam pengkodean dan menambahkan item secara dinamis ke tampilan antarmuka pengguna.

Aplikasi web telah menjadi bagian integral dalam kehidupan sehari-hari karena konten yang secara bebas diakses dari mana saja melalui internet (Deepa&Thilagam, 2016). Gambar di bawah ini menunjukkan ilustrasi arsitektur *three-tier* dari aplikasi web yang saat ini banyak digunakan beserta komponen *software* dan *hardware*.



Gambar 2.3 Arsitektur Aplikasi Web

(Sumber: Deepa&Thilagam, 2016)

### **Metodologi Pengembangan Aplikasi Web**

Langkah-langkah pengembangan aplikasi web dilakukan melalui metodologi. Menurut Dennis, et. al. (2015) metodologi pengembangan sebuah aplikias web merupakn pendekatan tersusu untuk melakukan SDLC (*system development life cycle*). SDLC memiliki empat fase dasar, yaitu perencanaan, analisis, desain, dan implementasi.

Jenis-jenis metodologi pengembangan aplikasi web antara lain:

***Waterfall***. Metodologi waterfall terdiri dari fase perencanaan, analisis, desain, implementasi dan perilisan aplikasi yang dilakukan secara bertahap dan berurutan. Kelebihan dari metode ini adalah proses identifikasi.

***Paralel***. Metodologi ini memiliki fase mulai dari perencanaan, analisis, dan pembuatan desain aplikasi secara menyeluruh yang dilakukan secara berurutan. Pada fase desain dan implementasi dibagi menjadi beberapa subproyek. Setelah semua subproyek selesai dilakukan kembali integrasi dan diikuti rilis aplikasi.

***Prototyping***. Metodologi ini memiliki fase yang mana siklus analisis, desain, dan implementasi dilakukan secara berulang sampai denga aplikasi tersebut komplit.

***Throwaway prototyping***. Metode ini hamper sama dengan prototyping, akan tetapi metode ini memiliki fase yang memerlukan analisis yang lebih teliti untuk mengembangkan konsep aplikasi.aplikasi dikembangkan menurut beberapa desain prototipe selamam fase analisis dan desain.tiap prototipe yang dibangun digunakan untuk meminimalkan resiko dengan mengidentifikasi adanya isu penting sebelum dibangun system yang sesungguhnya.

***Agile Development***. Metode ini berfokus pada penyederhanaan proses pengembangan aplikasi dengn mengeliminasi banyak biaya untuk membuat model dan dokumentasi serta waktu yang dibutuhkan utnuk melakukan fase tersebut.

### **Keamanan Informasi**

Merujuk ISO:IEC 27000, keamanan informasi adalah upaya yang dilakukan untuk menjaga kerahasiaan (*confidentiality*), integritas (*integrity*) dan ketersediaan (*availability*) dari informasi, hal ini dapat termasuk otentikasi (*authenticity*), akuntabilitas (*accountability*), nir-sangkal (*non-repudiation*) dan keandalan (*reliability*). Keamanan informasi mencakup keamanan dari level infrastruktur fisik, operasional, dan level kebijakan. Keamanan dalam dunia siber (*cybersecurity*) tercakup dalam konsep keamanan informasi. Berdasarkan ISO/IEC 27032:2012 *cybersecurity* adalah upaya yang dilakukan dalam menjaga kerahasiaan (*confidentiality*), integritas (*integrity*) dan ketersediaan (*availability*) dari informasi di dunia siber.

### **Keamanan Aplikasi Web**

Masalah keamanan aplikasi web biasanya berhubungan dengan masalah yang mungkin tidak disadari atau cenderung diabaikan oleh pengembang atau pemiliknya (Fajar, A & Yazid, S., 2018). Aplikasi web adalah salah satu target dalam serangan keamanan informasi. Banyak celah keamanan yang dapat dieksploitasi dapat menjadi sumber kerusakan dan gangguan operasional dari aplikasi itu sendiri dan bahkan infrastruktur TI di dalamnya.

### **Ancaman Keamanan Aplikasi Web**

Aplikasi web menjadi target utama ancaman keamanan dan serangan karena adanya *vulnerability* yang memicu serangan dan eksploitasi terhadap kerahasiaan, integritas, dan ketersediaan data (Deepa&Thilagam, 2016). Serangan pada aplikasi web menunjukkan kegagalan teknologi dalam mempertimbangkan keamanan aplikasi. Hal ini dipengaruhi oleh beberapa faktor berikut:

* + - 1. Terbatasnya dukungan keamanan mengakibatkan aplikasi web rentan terhadap serangan.
      2. Interoperabilitas dan keterbukaan XML yang menyediakan interaksi antar aplikasi web menjadi sasaran penyerang.
      3. Aplikasi web diimplementasikan oleh developer yang berfokus pada penerapan fitur dan fungsionalitas, tanpa mempertimbangkan aspek keamanan.

Pada Tahun 2017, OWASP (*Open Web Application Security Project*) mengeluarkan sebuah daftar ancaman yang dipublikasikan dalam dokumen OWASP Top 10. Berikut ini daftar sepuluh ancaman keamanan website yang ada pada OWASP Top 10 2017 (OWASP, 2017):

* *Injection*
* *Broken authentication*
* *Sensitive data exposure*
* *XML Eksternal Entities (XXE)*
* *Broken control access*
* *Security misconfiguration*
* *Cross-site scripting (XSS)*
* *Insecure deserialization*
* *Using component with known vulnerabilities*
* *Insufficient logging&monitoring*

### ***Vulnerability* Aplikasi Web**

Whitman (2012) menyatakan bahwa *vulnerability* adalah sebuah kelemahan yang teridentifikasi dalam sebuah aplikasi terkontrol, yang mana kontrolnya tidak ada atau tidak efektif lagi, sehingga ketika dieksploitasi akan menjadi potensi ancaman terhadap aplikasi. *Vulnerability* atau celah keamanan pada aplikasi web adalah suatu celah yang menjadi jalan penyerang untuk menyebabkan kerusakan pada aplikasi web. Jenis celah keamanan web yang pada umumnya dijumpai adalah sebagai berikut (Stuttard, 2011) :

***Input Validation Vulnerability.*** Kelemahahan yang paling umum dari aplikasi web adalah kegagalan untuk memvalidasi input yang berasal dari pengguna. Kelemahan ini menyebabkan adanya serangan-serangan terhadap aplikasi web seperti *SQL Injection*, *Cross Scripting* (XSS), *buffer overflow*, dan sebagainya

***Broken Authentication.*** Dalam keamanan komputer autentikasi adalah proses untuk memverifikasi indentitas digital dari pengirim dalam berkomunikasi. Kelemahan dalam kategori ini adalah berbagai cacat aplikasi dalam mekanisme login, seorang penyerang dapat dengan mudahnya untuk menebak password yang lemah, melancarkan serangan *brute force*, dan melewati login.

***Broken Access control/Authority Vulnerability.*** Dalam keamanan komputer, otorisasi adalah proses setelah autentikasi, untuk memverifikasi hak akses pengguna terhadap data. Kelemahan ini terjadi karena kegagalan aplikasi untuk melindungi akses ke data, sehingga memungkinkan penyerang untuk melihat data sensitif pengguna lain yang dimiliki server atau melakukan tindakan istimewa.

***Information leakage/Error Handling Vulnerability.*** Aspek penting dari pengembangan aplikasi yang aman adalah mencegah kebocoran informasi. Aplikasi membocorkan informasi sensitif yang berguna untuk penyerang dalam mengembangkan serangan selanjutnya terhadap aplikasi melalui pesan kesalahan. Aplikasi gagal karena menyajikan informasi penting ketika terjadi error.

***Session Management Vulnerability.*** Salah satu inti dari aplikasi web adalah mekanisme untuk mengontrol dan memanajemen status/keadaan pengguna ketika berinteraksi dengan aplikasi. Misalnya saat login bagaimana otentifikasi pengguna dilakukan dan bagaimana yang terjadi terhadap pengguna ketika mereka *log out*. Kelemahan dalam kategori ini berarti kegagalan aplikasi sehingga dapat digunakan pengguna dalam konteks tingkatan pengguna dan hak istimewa.

### **ISO/IEC 27001**

Achmadi, et.al. (2018) menyebutkan dalam penelitiannya bahwa ISO-IEC 27001 dapat digunakan untuk melakukan evaluasi terhadap aplikasi web. ISO-IEC 27001 menyediakan kerangka kerja privasi yang menentukan terminologi privasi umum; mendefinisikan para aktor dan peran mereka dalam memproses informasi yang dapat diidentifikasi secara pribadi (PII-*personal information identification*); menjelaskan pertimbangan pengamanan privasi; dan memberikan referensi ke prinsip privasi yang diketahui untuk teknologi informasi. ISO / IEC 27001 berlaku untuk orang perorangan dan organisasi yang terlibat dalam menentukan, mengadakan, merancang, mengembangkan, menguji, memelihara, mengelola, mengelola, serta mengoperasikan sistem atau layanan teknologi informasi dan komunikasi di mana kontrol privasi diperlukan untuk pemrosesan PII .

### **OWASP (*Open Web Application Security Project*)**

OWASP (*The Open Web Application Security Projek*) merupakan organisasi/komunitas yang fokus dibidang keamanan aplikasi. Semua kegiatan OWASP didedikasikan agar dapat membantu organisasi dalam memahami, mengembangkan, memperoleh, mengoperasikan dan merawat aplikasi yang aman atau dapat dipercaya. Semua hasil penelitian yang dilakukan oleh OWASP di sosialisasikan kepada siapapun yang ingin meningkatkan keamanan aplikasi dan semua itu bersifat terbuka baik berupa dokumen standar, tool, metodologi, artikel, serta teknologi. OWASP memiliki beragam dokumentasi, tools, dan framework yang digunakan untuk mendukung keamanan aplikasi web. Beberapa tools dan dokumentasi yang digunakan dalam penelitian ini dibahas dalam subbab-subbab berikut.

#### OWASP Top 10 Risk 2017

***Injection***. Serangan injection terjadi ketika terdapat data yang tidak terpercaya dikirim ke sebuah code interpreter melalui sebuah formulir input atau cara input data ke website lainnya. Injection ini bias berupa SQL, NoSQL, OS, dan LDAP. Data musuh yang diserang dapat mengelabuhi interpreter untuk mengeksekusi perintah yang tidak diinginkan atau mengakses data tanpa otorisasi yang tepat. *Vulnerability* ini dapat disebabkan oleh adanya imputasi yang tidak divalidasi oleh aplikasi atau adanya kueri dinamis pada interpreter.

***Broken Authentication***. Kelemahan pada sistem *login* bisa memberi peluang kepada penyerang utnuk melakukan akses ke akun user. Tidak hanya itu, penyerang bisa menguasai seluruh sistem dengan membajak akun admin. Untuk memitigasi kelemahan ini bisa digunakan *two-factor authentication* (2FA). Beberapa *vulnerability* yang dapat menyebabkan broken authentication antara lain:

* Mengizinkan serangan otomatis seperti isian kredensial, yang mana penyerang memiliki daftar nama pengguna dan kata sandi yang valid.
* Memungkinkan bruteforce atau serangan otomatis lainnya.
* Kata sandi yang standar atau lemah.
* Menampilkan ID sesi di URL.
* Tidak secara benar membatalkan ID sesi.

***Sensitive Data Exposure***. Sebuah aplikasi web yang menyimpan data-data sensitive penggunanya, akan berbahaya jika tidak dijaga keamanannya. Untuk mengurangi kemungkinan pencurian data, bisa dilakukan enkripsi terhadap data-data sensitif. Developer harus memastikan bahwa aplikasi web tidak menyimpan data-data sensitive yang sebenarnya tidak dibutuhkan.

***XML External Entities***. Ini adalah serangan ke website dan aplikasi yang menganalisa input XML. Input ini bisa mereferensikan entity external untuk mengetahui kelemahan yang ada pada input XMLnya. Entiti external yang dimaksud adalah berupa unit penyimpanan, misalnya hard drive. Analisa input XML bisa dibuat seolah-olah mengirim data ke entity external yang tidak dipercaya, sehingga data-data sensitive dapat diakses penyerang secara langsung. Cara terbaik untuk mengatasi XEE ini adalah dengan memiliki aplikasi web yang mengandung jenis data yang tidak terlalu kompleks.

***Broken Access Control***. *Access control* pada poin ini mengacu kepada sistem kontrol yang mengakses informasi dan fungsionalitasnya. *Access control* yang rusak memungkinkan penyerang untuk melewati proses autorisasi dan melakukan hal-hal yang biasanya hanya bisa dilakukan oleh admin. *Vulnerability* yang dapat menyebabkan *broken access control* adalah mengizinkan kunci utama untuk diubah ke catatan pengguna lain memungkinkan mellihat atau mengedit akun orang lain, dan adanya peningkatan hak istimewa, yaitu bertindak sebagai pengguna tanpa login atau sebagai admin ketika login sebagai pengguna.

***Security Misconfiguration***. Kesalahan konfigurasi keamanan adalah kelemahan yang paling sering terjadi di antara kelemahan lain di daftar ini. Biasanya kesalahan terjadi jika Anda hanya menggunakan konfigurasi *default* tanpa melihat kebutuhan website.

***Cross Site Scripting***. Kelemahan *cross site scripting* akan terjadi pada aplikasi web jika aplikasi tersebut menginjinkan pengguna untuk menambahkan kode *custom* ke sebuah path URL atau ke website yang dilihat oleh user lain. Kelemahan ini biasa dimanfaatkan untuk menjalankan kode JavaScript yang berbahaya pada browser korban. Contohnya, jika seorang penyerang mengirim email ke korban dengan mengatasnamakan nama bank tersebut dan menyertakan link ke website bank tersebut, mereka bisa saja menaruh kode JavaScript berbahaya ke dalamnya. Jika aplikasi web bank tidak terlindungi dengan baik, nasabah akan menjadi korban. Tiga bentuk XSS antara lain:

* *Reflected XSS*, penyerang dapat melakukan eksekusi HTML dan javascript pada browser pengguna.
* *Stored XSS*, aplikasi menyimpan input pengguna yang tidak terbersihkan sehingga dapat dilihat oleh pengguna lain di lain waktu.
* *DOM XSS*, sebuah kerangka javascript yang secara dinamis menyertakan data yang dapat dikontrol oleh penyerang ke suatu halaman yang rentan.

***Insecure Deserialization***. Serialisasi adalah proses dimana object diambil dari kode aplikasi dan diubah ke format lain sehingga bisa digunakan untuk keperluan lain, misalnya menyimpan data ke sebuah disk. Deserialisasi berarti sebaliknya; mengubah data yang sudah diserialisasi kembali object yang digunakan oleh aplikasinya. Insecure deserialization atau deserialisasi yang kurang aman bisa diserang dengan memanfaatkan data dari asal yang tidak dipercaya. Ini bisa menyebabkan terjadinya serangan DDoS. Untuk mencegah ini terjadi, perlu adanya larangan deserialisasi dari data yang tidak dipercaya.

***Using Components With Known Vulnerabilities***. Kebanyakan pada aplikasi web *developer* menggunakan komponen seperti *libraries* dan *frameworks*. Komponen-komponen ini adalah kumpulan *software* yang membantu *developers* untuk bekerja dengan lebih efisien. Beberapa hacker biasa mencari kelemahan yang ada pada komponen-komponen ini agar mereka bisa melakukan serangan. Oleh karena itu, *developer* harus selalu memastikan bahwa komponen-komponen ini sudah diupdate agar tetap aman.

***Insufficient Logging and Monitoring.*** Kebanyakan aplikasi web tidak mengambil langkah yang cukup untuk mendeteksi penembusan data. Rata-rata orang baru sadar saat terjadi penembusan di website mereka setelah 200 hari. Ini tentunya memberikan penyerang banyak waktu untuk melakukan penyerangan. OWASP merekomendasikan *developer* untuk mengimplementasi *logging* dan monitoring serta rencana response insiden agar mereka tahu jika ada penyerangan yang terjadi pada aplikasi mereka.

#### OWASP S-SDLC (*Secure software development lifecycle project*)

OWASP S-SDLC merupakan sebuah metode keamanan perangkat lunak menyeluruh untuk *developer* aplikasi web. Tujuannya adalah untuk mendefinisikan standar keamanan siklus pengembangan perangkat lunak (OWASP, 2018). Fase *lifecycle* OWASP S-SDLC terdiri dari:

1. Pelatihan

Bagian ini memberikan panduan pelatihan mengenai kesadaran keamanan. Pelatihan ini ditujukan untuk meningkatkan kepekaan *developer* terhadap keamanan dalam pengembangan perangkat lunak.

1. Persyaratan

Bagian ini bertujuan untuk memperoleh persyaratan keamanan dengan melakukan identifikasi implementasi fungsional, posisi dalam organisasi atau persyaratan keamanan umum.

1. Desain

Bagian ini memandu untuk memberikan desain keamanan yang dapat dilakukan oleh tim pelaksana dengan mempertimbangkan resiko eamanan teknis. Sehingga, dengan deteksi dini resiko keamanan, maka biaya untuk membangun aplikasi yang aman bias terkontrol.

1. Implementasi

Bagina ini bertujuan agar tim implementasi melakukan pengodean yang aman dan melakukan identifikasi dan perbaikan cacat dalam kode.

1. Pengujian

Pengujian keamanan adalah proses yang ditujukan untuk mengungkapkan kekurangan dalam mekanisme keamanan aplikasi yang melindungi data dan menjaga fungsionalitas sebagaimana dimaksud. Persyaratan umum dapat mencakupkerahasiaan, integritas, autentikasi, ketersediaan, otorisasi, dan *non-repudiation.*

1. *Deployment*

Bagian ini berfokus pada audit keamanan sebelum dilakukan *deployment* dan pemantauan keamanan. Yang perlu diperhatikan adalah pengembangan *baseline* keamanan yang sesuai dengan *deployment*, dan proses tanggapan terhadap insiden dan teknologi terkait.

#### OWASP ASVS (*Application Security Verification Standars*)

OWASP ASVS adalah sebuah daftar persyaratan bagi *developer* dalam menentukan apakah sebuah aplikasi itu aman untuk digunakan oleh organisasi atau tidak. OWASP ASVS juga berisi dasar untuk menguji kontrol keamanan teknis aplikasi web serta kontrol keamanan apapun di lingkungan pengembangan untuk melindungi aplikasi terhadap *vulnerability*. Versi ASVS terakhir adalah versi 4.0 (OWASP, 2019).

Tujuan utama ASVS adalah:

* + - 1. Sebagai metrik, yang memberikan tolok ukur bagi pengembang dan pemilik aplikasi web dalam menilai tingkat kepercayaan.
      2. Sebagai panduan, yang memberikan panduan pengembangan kontrol keamanan bagi developer agar aplikasi dapat memenuhi persyaratan kemanan.
      3. Sebagai dasar dalam pengadaan atau akuisisi untuk menentukan persyaratan verifikasi keamanan aplikasi dalam kontrak.

ASVS dapat diaplikasikan ke dalam kontrol aplikasi yang dibagi ke dalam 3 level verifikasi keamanan, yaitu:

1. Level 1, merupakan verifikasi langkah pertama dan syarat minimum untuk aplikasi secara umum.
2. Level 2, merupakan verifikasi untuk aplikasi-aplikasi yang berisi data sensitive dan membutuhkan perlindungan, serta aplikasi yang terkait dengan integritas bisnis sutu organisasi.
3. Level 3, merupakan verifikasi tertinggi untuk aplikasi-aplikasi paling penting atau aplikasi kritis yang bernilai transaksi tinggi, dan data sensitive, dimana jika terjadi kegagalan akan berdampak terhadap operasi organisasi.

ASVS terdiri atas 14 poin persyaratan aplikasi web yang meliputi:

1. Persyaratan rsitektur, desain, dan pemodelan ancaman
2. Persyaratan verifikasi autentikasi
3. Persyaratan manajemen sesi
4. Persyaratan verifikasi kontrol akses
5. Persyaratan verifikasi validasi, sanitasi dan *encoding*
6. Persyaratan verifikasi kriptografi tersimpan
7. Persyaratan verifikasi penanganan kesalahan dan pencatatan log
8. Persyaratan verifikasi perlindungan data
9. Persyaratan verifikasi komunikasi data
10. Persyaratan verifikasi *malicious code*
11. Persyaratan verifikasi logika bisnis
12. Persyaratan verifikasi file dan sumber daya
13. Persyaratan verifikasi API dan web servis
14. Persyaratan verifikasi konfigurasi

#### OWASP Cheat Sheet 2015

OWASP *cheat sheet* versi 2015 merupakan pedoman untuk *developer* dalam menilai keamanan sebuah aplikasi web dengan lebih detail dan mendalam. *Cheat sheet* ini dapat membantu *developer* dengan keahlian khusus dalam topik tertentu seperti kriteria keamanan autentikasi, pengujian penetrasi, dan lain-lain.

#### OWASP *Testing Guide*

OWASP *testing guide* v5 ini merupakan panduan standar untuk melakukan pengujian penetrasi pada aplikasi web. Panduan ini dapat membantu pengembangan aplikasi web yang merupakan kombinasi orang, proses, dan teknologi (OWASP, 2019). Langkah-langkah pengujian keamanan web meliputi:

1. Pengunpulan informasi
2. Pengujian manajemen konfigurasi dan deployment
3. Pengujian manajemen identitas
4. Pengujian autentikasi
5. Pengujian otorisasi
6. Pengujian manajemen sesi
7. Pengujian validasi data
8. Pengujian penanganan kesalahan
9. Pengujiank kriptografi
10. Pengujian logika bisnis
11. Pengujian pada sisi klien

## **Penelitian Sebelumnya**

**Fajar dan Yazid (2018)** mengadakan penelitian mengenai keamanan aplikasi web dengan judul *Enumeration Enumeration and Handling Security Issues of Government Official Web Application.* Penelitian ini bertujuan untuk mengidentifikasi isu keamanan yang ada pada aplikasi web resmi pemerintahan. Penelitian ini dilatarbelakangi adanya temuan pada aplikasi web di pemerintah yang banyak mengandung *vulnerability*. Dengan menggunakan Framework OWASP, Fajar dan Yazid melakukan uji vulnerability pada beberapa web pemerintah daerah dengan tiga alat untuk *penetration testing* salah satunya adalah aplikasi OWASP ZAP. Hasil peneilitian ini didapatkan kesimpulan bahwa vulnerabilitas pada aplikasi web sering disebabkan karena masalah konfigurasi pada aplikasi dan keamanan. Masalah tersebut diakibatkan karena ketika melakukan perancangan pada pengembangan aplikasi mengabaikan kebutuhan keamanan. Penelitian ini juga menyebutkan bahwa OWASP ZAP dapat memberikan akurasi yang lebih baik dalam melakukan uji penetrasi dibanding alat yang lain. Pada akhir penelitian, Fajar dan Yazid memberikan rekomendasi langkah-langkah untuk membuat aplikasi lebih aman yang mencakup sebagai berikut (1) Pengembangan Aplikasi Aman, (2) Lingkungan Aplikasi dan (3) Tata Kelola Keamanan Aplikasi.

**Tatli, (2018)** mengadakan penelitian mengenai keamanan aplikasi web dengan judul *Developer-oriented Web Security by Integrating Secure SDLC into IDEs*. Penelitian ini melakukan analisis terhadap beberapa metodologi terkait SSDLC (*Secure Software Development Life Cycle*) yaitu Microsoft SDL, OpenSAMM, BSIMM, dan OWASP SSDLC. Peneliti melakukan integrasi metodologi-metodologi tersebut dan mengaplikasikan dalam sebuah IDE (*Integrated Development Environment*). Developer dapat dengan mudah mengaplikasikan keamanan dalam pembangunan aplikasi web karena langsung melakukan akses pada IDE tersebut. Hal ini sangat signifikan dalam mengurangi jumlah vulnerability dan mencegah kebocoran data penting.

**Achmadi, et.al. (2018)** melakukan penelitian mengenai keamanan informasi dengan judul *On developing information security management system (ISMS) framework for ISO 27001-based data center*. Penelitian ini bertujuan untuk menyusun kerangka kerja ISMS khusus untuk pusat data untuk mengelola aspek kerahasiaan, integritas, dan ketersediaan keamanan informasi berdasarkan ISO/IEC 27001.

**Sedek, et.al. (2009)** mengadakan penelitian mengenai keamanan aplikasi web dengan judul *Developing a Secure Web Application Using OWASP Guidelines*. Penelitian ini melakukan evaluasi bagaimana OWASP dapat membantu developer dalam membangun web yang aman dengan tujuan untuk membangun sebuah panduan dalam pembangunan web yang aman bagi developer berdasarkan OWASP *Guidelines*. Dokumen OWASP yang digunakan untuk membangun pedoman keamanan pembangunan aplikasi web adalah OWASP SDLC, OWASP TOP 10 Vulnerability, OWASP ASVS. Hasil penelitian menunjukkan bahwa framework OWASP dapat 100% dapat membantu developer dalam membangun aplikasi berbasis web yang aman setidaknya dari 10 kerentanan paling kritis.*.*

Tabel 2.1 Perbandingan dengan Penelitian-penelitian Sebelumnya

| **Komponen analisis** | **Fajar A.& Yazid S. (2018)** | **Tatli (2018)** | **Achmadi (2018)** | **Sedek (2018)** |
| --- | --- | --- | --- | --- |
| Judul | ***Enumeration and Handling Security Issues of Government Official Web Application*** | ***Developer-oriented Web Security by Integrating Secure SDLC into IDEs*** | ***On Developing Information Security Management System (ISMS) Framework for ISO 27001-based Data Center*** | ***Developing a Secure Web Application Using OWASP Guidelines*** |
| Masalah | Web di pemerintahan banyak mengandung vulerabilitas | Developer mengalami kesulitan dalam mengadopsi kebutuhan keamanan dalam pengembangan aplikasi web | Belum ada standar keamanan yang spesifik untuk data center | Developer memerlukan panduan untuk membangun aplikasi web yang aman. |
| Tujuan | Untuk mengidentifikasi isu keamanan yang ada pada aplikasi web resmi pemerintahan | melakukan analisis terhadap beberapa metodologi terkait SSDLC dengan melakukan integrasi metodologi-metodologi tersebut dan mengaplikasikan dalam sebuah IDE (Integrated Development Environment) | Menyusun kerangka kerja ISMS khusus untuk pusat data untuk mengelola aspek kerahasiaan, integritas, dan ketersediaan keamanan informasi berdasarkan ISO 27001 | Mengevaluasi bagaimana OWASP dapat membantu developer dalam membangun aplikasi web yang aman. |
| Metodologi | *Experimental dan case study* | *Experimental dan case study* | *Experimental dan case study* | Experimental dan case study |
| Temuan/Hasil | aplikasi web memiliki kelemahan paling dalam masalah konfigurasi, baik konfigurasi aplikasi maupun keamanan | dibangun sebuah IDE yang mengimplementasikan keamanan aplikasi web. | dibentuk sebuah framework keamanan berdasarkan ISO 27001 sehingga dapat diimplementasikan dalam pembangunan paplikasi web pada data center | Dengan hasil 100% ditemukan bahwa OWASP sangat membantu dalam membangun aplikasi berbasis web yang aman setidaknya dari 10 kerentanan paling kritis |
| Kesimpulan | Semua kelemahan aplikasi web yang diidentifikasi berkaitan dengan lingkungan aplikasi dan dukungan pengembangan teknis, pengembangan aplikasi web sering mengabaikan persyaratan keamanan selama desain, pengembangan, dan penyebaran | Developer aplikasi web dapat dipermudah dengan penerapan IDE untuk mengadopsi kebutuhan keamanan. | Standar ISO 27001 dapat dijadikan standar untuk membuat framework keamanan informasi. | OWASP merupakan salah satu framework keamanan yang dapat membantu developer dalam membangun aplikasi web yang aman. |
| Batasan | evaluasi hanya dilakukan pada web pemerintah daerah dan tidak disebutkan nama web yang dievaluasi. | IDE yang dibangun hanya sampai batasan konsep | Framework yang dibangun hanya adopsi standar ISO 27001 | Kriteria yang dibahas dari OWASP hanya terkait 10 kerentanan/vulnerabilitas kritis. |
| Relevansi | Penelitian ini relevan dalam hal penggunaan framework OWASP, dan dilakukan terhadap website di pemerintahan | Penelitian ini relevan dalam hal penggunaan metodologi OWASP SSDLC dalam implementasi keamanan aplikasi web | Penelitian ini relevan dalam hal penggunaan ISO 27001 sebagai salah satu acuan pembuatan framework. | Penelitian ini relevan dalam mendukung penelitian yang akan dilakukan dalam penggunaan OWASP sebagai framework |
| *Compare* | Penelitian hanya berfokus kepada evaluasi aplikasi web pemerintahan tanpa melakukan pembuatan prosedur keamanan pengembangan aplikasi web | Penelitian ini hanya berfokus pada pembuatan pembangunan IDE tidak membuat prosedur keamanan pada pengembangan aplikasi. | Penelitian ini hanya berfokus pada pembuatan framework keamanan informasi tidak membuat prosedur keamanan pada pengembangan aplikasi | Sama-sama membahas megenai keamanan aplikasi web dan penggunaanframework OWASP. |
| *Contrast* | Sampel web yang dievaluasi tidak didefinisikan dan tidak dijelaskan | Tujuan penelitian lebih membahas tentang keamanan pada aplikasi web terkait data privasi. | Metode penelitian yang digunakan berbeda yaitu experimental dengan uji komparasi terhadap performa 2 scanner vulnerabilitas aplikasi web | Penelitian ini berfokus pada evaluasi framework OWASP sebagai panduan bagi developer dalam membangun website yang aman. |
| *Critisize* | Semua kelemahan yang diidentifikasi berkaitan dengan lingkungan aplikasi dan dukungan pengembangan teknis, pengembangan aplikasi web sering mengabaikan persyaratan keamanan selama desain, pengembangan, dan penyebaran | Wawancara hanya dilakukan pada 1 responden | Pengujian hanya dilakukan terhadap 4 kategori vulrabilitas | Evaluasi dokumen OWASP lebih baik jika dilakukan tidak hanya pada OWASP top 10 |
| *Summarize* | Untuk melakukan evaluasi terhadap keamanan aplikasi web dapat menggunakan tools pent-test untuk mendeteksi vulnerabilitas, framework OWASP untuk panduan testing aplikasi web | Diperlukan adopsi dan dokumentasi kebijakan dan prosedur mengenai keamanan informasi secara komprehensif. | OWASP memiliki performa lebih baik daripada WAVSEP | OWASP merupakan salah satu framework keamanan yang dapat membantu developer dalam membangun aplikasi web yang aman. |
| *Syntesize* | Penelitian hanya berfokus kepada evaluasi aplikasi web pemerintahan tanpa melakukan pembuatan prosedur keamanan pengembangan aplikasi web | Perlu diterapkan kebijakan dan prosedur keamanan terkait keamanan informasi secara komprehensif. | Komparasi tools scanner vulnerabilitas dapat dijadikan rujukan dalam melakukan pemilihan tools untuk dijadikan bencmark | OWASP dapat dijadika acuan bagi developer dalam membangun aplikasi web yang aman. |

## **Metodologi**

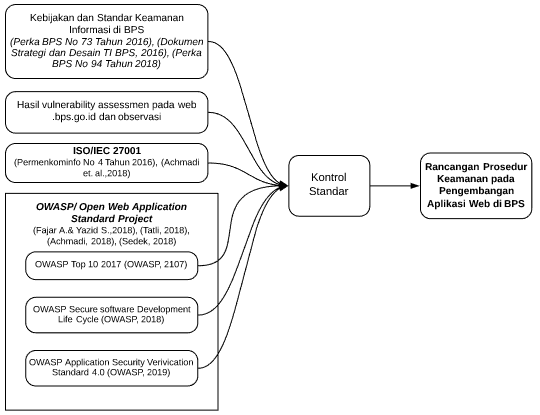
Penelitian ini menggunakan metode penelitian *case study* dengan melakukan analisis terhadap data dan dokumentasi secara kualitatif dan didukung dengan wawancara terhadap ahli dan praktisi pada bidang terkait keamanan di Badan Pusat Statistik. Framework yang akan digunakan adalah menggunakan OWASP. Untuk itu peneliti melakukan studi komparasi terhadap penelitian-penelitian terdahulu untuk mengidentifikasi keterkaitan dan kesamaan sebagai acuan dalam menunjang penelitian untuk mencapai tujuan. Berikut table komparasi penelitian-penelitian terdahulu terkait framework dan metodologi.

Tabel 2.2 Perbandingan Framework dan Metodologi Olah Data

| **No** | **3C+2S** | | **Fajar A.& Yazid S. (2018)** | **Tatli (2018)** | **Achmadi (2018)** | **Sedek (2018)** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | *Compare* | |  |  |  |  |
| 2 | *Contrast* | Framework | OWASP | Microsoft SDL, OpenSAMM, BSIMM, dan OWASP SSDLC |  | OWASP |
| Metode penelitian | Case Study | Case Study | Case Study | Case study |
| Metode koleksi data | Observasi, studi literatur | Observasi, studi literatur | Observasi, studi literatur | Observasi, studi literatur |
| Metode olah data | Analisis deskriptif | Analisis deskriptif | Analisis deskriptif | Analisis deskriptif |
| 3 | *Criticize* | | Tidak dijelaskan sampel website yang dilakukan evaluasi | Tidak dijelaskan mengenai pemilihan framework. | Tidak dijelaskan mengenai pemilihan framework. | Evaluasi dokumen OWASP lebih baik jika dilakukan tidak hanya pada OWASP top 10 |
| 4 | *Sytesize* | | Untuk melakukan evaluasi terhadap keamanan aplikasi web dapat menggunakan tools pent-test untuk mendeteksi vulnerabilitas, framework OWASP untuk panduan testing aplikasi web | Untuk meningkatkan kesadaran keamanan informasi dapat dilakukan adopsi kebijakan/prosedur secara komprehensif | Untuk membuat framework keamanan aplikasi web dapat menggunakan standar ISO/IEC 27001 | OWASP dapat dijadikan acuan bagi developer dalam membangun aplikasi web yang aman. |
| 5 | *Summarize* | | Semua kelemahan yang diidentifikasi berkaitan dengan lingkungan aplikasi dan dukungan pengembangan teknis, pengembangan aplikasi web sering mengabaikan persyaratan keamanan selama desain, pengembangan, dan penyebaran | Perlu diterapkan kebijakan dan prosedur keamanan terkait keamanan informasi secara komprehensif. | Kerangka kerja dapat menggunakan OWASP pada pengujian efektifitas test | OWASP merupakan salah satu framework keamanan yang dapat membantu developer dalam membangun aplikasi web yang aman. |

## ***Theoretical Framework***

Dari landasan teori dan penelitian sebelumnya, penulis membuat *theoretical framework* sebagai berikut:



Gambar 2.4 Theoretical Framework

Pembuatan kontrol standar yang menghasilkan sebuah rancangan prosedur keamanan pada pengembangan aplikasi web di BPS ddilakukan berdasarkan komposisi dari kententuan instansi dan hasil *vulnerability assessment* terhadap web .bps.go.id pada tahun 2018 serta penelitian terdahulu.

1. Kebijakan dan standar keamanan informasi di BPS. Dokumen yang dijadikan acuan adalah Perka BPS No 73 Tahun 2016, Dokumen Strategi dan Desain TI BPS Tahu 2016, dan Perka BPS No 94 Tahun 2018. Dokumen-dokumen ini yang akan menjadi kerangka umum dari butir-butir prosedur keamanan pengembangan aplikasi web.
2. Hasil asesmen *vulnerability* pada web .bps.go.id dan observasi developer. Hasil asesmen dan observasi ini akan memberikan informasi berupa poin-poin vulnerability yang perlu dibuat solusinya. Hasil observasi akan memperkuat pembuatan poin-poin vulnerability hasil assesmen.
3. Standar ISO/IEC 27001. Dokumen yang dijadikan acuan adalah ISO/IEC 27001:2013 berdasarkan penelitian dari Achmadi et. el. (2018) dan Peraturan Menteri Komunikasi dan Informasi No 4 Tahun 2016 tentang sistem manajemen pengamanan informasi. Standar ini dijadikan acuan mengikuti peraturan tersebut.
4. Standar OWASP (*Open Web Application Standard Project*). Dokumen OWASP yang dijadikan acuan adalah OWASP TOP 10 2017, OWASP SSDLC, dan OWASP ASVS berdasarkan penelitian Fajar A.& Yazid S. (2018), Tatli, (2018), dan Sedek, (2018). OWASP TOP 10 2017 akan menjadi kerangka asesmen menggunakan alat uji penetrasi menggunakan OWASP ZAP untuk menilai tingkat kritis pada 10 jenis vulnerability. OWASP SSDLC akan menjadi kerangka dalam membuat pedoman urutan-urutan siklus hidup dari pengembangan aplikasi. OWASP ASVS menjadi kerangka penjabaran dari kerangka utama sesuai dengan hasil asesmen.

# **METODOLOGI PENELITIAN**

Bab ini berisi desain penelitin, alur penelitian, pengumpulan data dan penyusunan rekomendasi.

## **Desain Penelitian**

Penelitian ini menggunakan metode penelitian yang bersifat kuantitatif dan kualitatif (*mixed-method*) dengan melakukan *case study* pada sebuah instansi. Metode Kulantitatif digunakan untuk mengukur tingkat dan jenis vulnerability pada aplikasi web di BPS. Metode kualitatif dirancang untuk membantu peneliti dalam memahami fenomena dalam konteks (Recker, 2013). Pada penelitian ini *case study* dilakukan pada sebuah instansi pemerintahan yaitu Badan Pusat Statistik. Menurut kegunaannya, penelitian ini dikategorikan juga sebagai *applied research*, karena penelitian ini memiliki tujuan untuk menerapkan hasil temuan penelitian lain untuk digunakan kembali dalam pemecahan permasalahan organisasi tertentu (Sekaran, 2013).

Metode pengumpulan data dilakukan dengan teknik observasi yaitu observasi pada aplikasi web dan diikuti observasi pada developer aplikasi web yang bersangkutan. Selain observasi juga dilakukan studi literatur terhadap sejumlah dokumen yang disebutkan dalam kerangka teoretis. Variabel observasi mengikuti daftar dalam OWASP Top 10 vulnerability untuk observasi aplikasi web, dan daftar 14 kontrol keamanan pada OWASP ASVS unutk observasi developer. Observasi tersebut menggunakan alat berupa OWASP ZAP untuk uji penetrasi, dan lembar observasi developer seperti contoh pada Lampiran 11.

Data yang telah diperoleh dari hasil pengumpulan data yang berupa data kualitatif dan kuantitatif kemudian diolah dengan cara melakukan akumulasi dan kodifikasi. Tahapan-tahapan lebih detail dijelaskan pada poin 3.2 alur penelitian.

Desain penetian secara ringkas dijelaskan pada Tabel 3.1 *Desain Penelitian* berikut.

Tabel 3.1 Desain Penelitian

| **Elemen** | **Keterangan** |
| --- | --- |
| Kualifikasi | *Case study, Mixed-method* |
| Paradigm | *Positivism* |
| Tujuan Penelitian | Rekomendasi rancangan prosedur keamanan pada pengembangan aplikasi web |
| Jenis Data | Primer: kuantitatif dan kualitatif |
| Metode Pengumpulan Data | * Studi dokumen: Dokumen-dokumen pada kerangka teoretis * Observasi: aplikasi web dan developer |
| Unit Observasi | * aplikasi web layanan data pada server hosting di Pusat Data BPS * 5 *developer* |
| Variabel Observasi | * Daftar OWASP *Top* 10 *Vulnerability* * Daftar OWASP ASVS 14 kontrol keamanan aplikasi |
| Instrumen Penelitian | * OWASP-ZAP *Penetration Testing* * Lembar Observasi Developer (Lampiran 11) |
| Metode Olah Data | Akumulasi hasil observasi dan kodifikasi studi dokumen |
| Metode Analisis Data | Kuantitatif dan Kualitatif, Deskriptif |
| Metode Penarikan Kesimpulan | Induktif |

## **Alur Penelitian**

Alur penelitian menggambarkan tahapan-tahapan yang dilakukan dalam penelitian. Penelitian ini dibagi menjadi 7 tahap dimulai dari pengumpulan data awal, perumusan masalah sampai dengan penyusunan kesimpulan penelitian. Secara detail tahapan penelitian beserta input, proses, dan output dapat dilihat pada Gambar 3.1 Tahapan Penelitian.

|  |  |  |
| --- | --- | --- |
| Tahap 1 Pengumpulan Data Awal | | |
| Input | **Proses** | **Output** |
| Log insiden  Hasil tes *vulnerability* | Studi dokumen, wawancara | Gap antara harapan dan kenyataan |

|  |  |  |
| --- | --- | --- |
| Tahap 2 Perumusan Masalah | | |
| Input | **Proses** | **Output** |
| Gap antara harapan dan kenyataan | *Gap analysis*, Studi dokumen, wawancara | *Research Question* |

|  |  |  |
| --- | --- | --- |
| Tahap 3 Tinjauan Pustaka | | |
| Input | **Proses** | **Output** |
| *Research Question* | Studi literatur menggunakan metode 3C+2S | *Theoretical Framework* |

|  |  |  |
| --- | --- | --- |
| Tahap 4 Perumusan Metodologi Penelitian | | |
| Input | **Proses** | **Output** |
| *Theoretical Framework* | Studi literatur | Metodologi penelitian |

|  |  |  |
| --- | --- | --- |
| Tahap 5 Penyusunan Instrumen Penelitian | | |
| Input | **Proses** | **Output** |
| *Metodologi Penelitian* | Studi dokumen OWASP Top 10 2017, OWASP ASVS 4.0 | Lembar observasi developer, Daftar top 10 *vulnerability* aplikasi web |

|  |  |  |
| --- | --- | --- |
| Tahap 6 Pengumpulan data lanjutan | | |
| Input | **Proses** | **Output** |
| Lembar observasi developer, Daftar top 10 *vulnerability* aplikasi web | Observasi, *penetration test/pentest* | lembar observasi yang telah diisi developer, laporan hasil *pentest* |

|  |  |  |
| --- | --- | --- |
| Tahap 7 Pengolahan dan analisis data hasil pengumpulan | | |
| Input | **Proses** | **Output** |
| lembar observasi yang telah diisi developer, laporan hasil *pentest* | Rekapitulasi hasil isian lembar observasi dan hasil pentest | Dokumen tentang statistik *vulnerability* dan kondisi (level) aplikasi web saat ini |

|  |  |  |
| --- | --- | --- |
| Tahap 8 Penyusunan Rekomendasi rancangan prosedur | | |
| Input | **Proses** | **Output** |
| Dokumen tentang statistik *vulnerability* dan kondisi (level) aplikasi web saat ini | analisis deskriptif dan pemetaan terhadap kondisi saat ini dengan standar OWASP ASVS | Dokumen rekomendasi rancangan prosedur keamanan pada pengembangan aplikasi web |

|  |  |  |
| --- | --- | --- |
| Tahap 9 Validasi hasil penyusunan rekomendasi rancangan prosedur | | |
| Input | **Proses** | **Output** |
| Rekomendasi rancangan prosedur keamanan pada pengembangan aplikasi web | Konfirmasi kepada pengelola aplikasi web di BPS | Rancangan prosedur keamanan pada pengembangan aplikasi web |

Gambar 3.1 Tahapan Penelitian

## **Pengumpulan Data**

Secara umum penelitian ini menggunakan metode pengumpulan data berupa observasi (asesmen) dan studi dokumen. Teknik observasi dilakukan dalam dua tahapan yaitu observasi pada aplikasi web dan diikuti observasi pada developer aplikasi web yang bersangkutan. Selain observasi juga dilakukan studi literatur terhadap sejumlah dokumen yang disebutkan dalam kerangka teoretis. Variabel observasi mengikuti daftar dalam OWASP Top 10 vulnerability untuk observasi aplikasi web, dan daftar 14 kontrol keamanan pada OWASP ASVS unutk observasi developer. Observasi tersebut menggunakan alat berupa OWASP ZAP untuk uji penetrasi, dan lembar observasi developer seperti contoh pada *Lampiran 11.*

## **Penyusunan Rekomendasi**

Penyusunan rekomendasi rancangan prosedur keamanan pada pengembangan aplikasi web dilakukan dengan melakukan studi literatur terhadap dokumen-dokumen berikut:

1. Kebijakan dan standar keamanan informasi di BPS yaitu Perka BPS No 73 Tahun 2016, Dokumen Strategi dan Desain TI BPS Tahu 2016, dan Perka BPS No 94 Tahun 2018. Dokumen-dokumen ini yang akan menjadi kerangka umum dari butir-butir prosedur keamanan pengembangan aplikasi web.
2. Dokumen Standar ISO/IEC 27001.
3. Dokumen Standar OWASP (*Open Web Application Standard Project*) meliputi OWASP TOP 10 2017, OWASP SSDLC, dan OWASP ASVS. OWASP TOP 10 2017 akan menjadi kerangka asesmen menggunakan alat uji penetrasi menggunakan OWASP ZAP untuk menilai tingkat kritis pada 10 jenis vulnerability. OWASP SSDLC akan menjadi kerangka dalam membuat pedoman urutan-urutan siklus hidup dari pengembangan aplikasi. OWASP ASVS menjadi kerangka penjabaran dari kerangka utama sesuai dengan hasil asesmen

Akumulasi hasil obeservasi dan studi literatur menghasilkan sebuah draf rancangan prosedur keamanan pada pengembagan aplikasi web. Draf yang telah dilakukan konfirmasi dan disetujui oelh pengelola aplikasi web di BPS menghasilkan rancangan final prosedur keamanan pada pengembangan aplikasi web yang siap diimplementasikan.

## **Jadwal Penelitian**

Penelitian akan dilakukan menurut jadwal berikut:



Tahapan pada jadwal di atas mengikuti tahapan pada poin 3.2. Penelitian telah dilakukan sampai tahap pengumpulan instrumen penelitian. Pada bulan Juni 2019 samapi dengan Desember 2019 akan dilanjutkan tahapan penelitian berikutnya.

## **DAFTAR PUSTAKA**

A. C. Nwaeze, P. Zavarsky, and R. Ruhl, *Compliance evaluation of information privacy protection in e-government systems in Anglophone West Africa using ISO/IEC 29100:2011*, (2017). 12th Int. Conf. Digit. Inf. Manag. ICDIM 2017, vol. 2018–January, no. Icdim, pp. 98–102, 2018.

A. Nguyen-tuong, S. Guarnieri, D. Greene, J. Shirley, and D. Evans, (2015). *Automatically Hardening Web Applications Using Precise Tainting.*

Alsmadi, I. (2016). *E-government website security concerns and citizens ’ adoption*. vol. 12, no. 3, pp. 243–255.

Alsmadi, I. & Mira, F. (2018). *Website security analysis : variation of detection methods and decisions*. 21st Saudi Comput. Soc. Natl. Comput. Conf., pp. 1–5.

B. Mburano & S. W. Si, “Evaluation of Web Vulnerability Scanners Based on OWASP Benchmark” . Western Sydney University,” no. June, 2018

BPS. (2015). *Rencana strategis (Renstra) 2015-2019 Badan Pusat Statistik (BPS) RI*. Jakarta: BPS.

BPS. (2016). *Dokumen Strategi dan Desain TI BPS*. Jakarta: BPS

BPS. (2018). *Laporan Kinerja Direktorat SIS 2018*. Jakarta: BPS

BPS. (2016). *Perka BPS No 73 Tahun 2016 Tentang Prinsip Tata Kelola TI di BPS*. Jakarta: BPS

BPS. (2018). *Perka BPS No 94 Tahun 2018 Tentang rencana transformasi digital BPS*. Jakarta: BPS

D. Achmadi, Y. Suryanto, and K. Ramli, “On developing information security management system (isms) framework for iso 27001-based data center,” 2018 Int. Work. Big Data Inf. Secur. IWBIS 2018, pp. 149–157, 2018.

D. Cannon (2011). ITIL Service Strategy Second Edition. London : The Stationary Office.

D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, and T. P. Group, “Preferred Reporting Items for Systematic Reviews and Meta-Analyses : The PRISMA Statement,” vol. 6, no. 7, 2009.

Dennis, A., Wixom, B.H., &Tegarden, D. (2015). *System Analysis and design an object oriented approach with uml 5th edition.* John Wiley Son.

E. İ. Tatli, *Developer-oriented Web Security by Integrating Secure SDLC into IDEs*, vol. 1, no. April, 2018.

Fajar, A & Yazid, S. (2018). *Enumeration and handling security issues of government official web application*. Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2018, no. 2014, pp. 81–86, 2019.

G. Alazman, M. Alabdali, D. Almubayedh, R. Al-Refai, M. Al khalis, and N. Nagy, *Security Related Issues In Saudi Arabia Small Organizations: A Saudi Case Study*, (2018). 21st Saudi Comput. Soc. Natl. Comput. Conf., pp. 1–6, 2018.

G. Deepa and P. S. Thilagam, *Securing web applications from injection and logic vulnerabilities: Approaches and challenges*, Inf. Softw. Technol., vol. 74, pp. 160–180, 2016.

J. Molina, M. Zea, J. Honores, and A. Gómez, *Analysis Methodologies Web Application Development*, Int. J. Appl. Eng. Res., vol. 11, no. 0973-4562, pp. 9070–9078, 2016.

OISSG. (2006). *Information System Security assessment Framework*. OISSG Group.

OWASP. (2016). *OWASP Application Security Verification Standard*. OWASP.

OWASP. (2019). *OWASP Testing Guide*. OWASP.

OWASP. (2018). *OWASP Top 10-2017*. OWASP.

OWASP. (2018). *OWASP SSDLC*. OWASP.

OWASP. (2015). *OWASP Cheat sheet*. OWASP.

Peraturan Pemerintah Republik Indonesia Nomor 82 Tahun 2012.

Pereturan Menteri Komunikasi dan Informasi Nomor 4 Tahun 2016.

Recker, Jan. (2013). Scientific Research in Information Systems: A Beginner’s Guide. New York: Springer Heidelberg.

Sekaran, U. (2013). Research Methods For Business: A Skill Building Approach (6th Edition). New York: John Wiley & Sons.

Stuttard. (2011). *Web Application Hacker's Handbook (2nd, 12)*. Dafydd - Pinto, Marcus.

T. Van Goethem, P. Chen, N. Nikiforakis, L. Desmet, and W. Joosen, *Large-scale security analysis of the web: Challenges and findings,* (2014). Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8564 LNCS, pp. 110–126, 2014.

UU-ITE *UU Nomor 11 Tahun 2008*

Van Os. R., (2016). *SOC-CMM: Designing and Evaluating a Tool for Measurement of Capability Maturity in Security Operations Centers*, p. 74.

Whitman, M. E., & Mattord, H. J., (2012). *Principle of Information Security*. BostonL Course Technology

Y. Yu, Y. Yang, J. Gu, and L. Shen, *Analysis and suggestions for the security of web applications*, Proc. 2011 Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2011, vol. 1, pp. 236–240, 2011.

https://bssn.go.id/bssn-lakukan-drill-test-penanggulangan-insiden-web-defacement-sektor-pemerintah

## **LAMPIRAN**

**Lampiran 1** Transkrip Wawancara dengan Kepala Subdirektorat Jaringan Komunikasi Data

Nama Narasumber: Agung Gumilar Triyanto

Jabatan: Kepala Subdirektorat Integrasi Pengolahan Data Statistik Sosial

Hari/Tanggal: Senin / 25 Februari 2019

Kode: W0

ND: Nia Dwi Rahayuningtyas

AG: Agung Gumilar Triyanto

ND : Seperti yang disebutkan dalam laporan, menurut Bapak apakah yang menjadi penyebab utama terjadinya serangan website di BPS?

AG : Lebih disebabkan oleh system dan aplikasi, karena banyaknya *vulnerability* pada aplikasi, sehingga serangan terutama *file injection* gampang masuknya (W0-01). Kalau sudah masuk ke server akan mudah sekali menjangkit ke website lain di server yang sama.

ND : Apakah sudah ada langkah yang dilakukan untuk menangani insiden tersebut ?

AG : sudah ada

ND : Apa saja Pak langkah-langkahnya ?

AG : Pertama kita lakukan upgrade web, kemudian *upgrade script engine* juga. Kita juga berkomunikasi dengan *developer* aplikasi website untuk perbaikan aplikasi (W0-02).

ND : Apakah sudah ada kebijakan mengenai mekanisme/prosedur/langkah deteksi, prevensi, dan penanganan insiden serangan website ?

AG : Kebijakan tertulis, yang *legal standing* misalnya dalam bentuk Perka/Kepka belum ada (W0-03). Sedangkan untuk deteksi, prevensi, dan penanganan berdasarkan koordinasi dan kesepakatan di level teknis (W0-04).

ND : Apa saja kendala yang dihadapi dalam penanganan insiden tersebut?

AG : SDLC belum diterapkan dengan benar, kesadaran yang belum merata, kemudian urgensi aplikasi timelinenya pendek, akibatnya pengembangan dan pengujian aplikasi kurang memadai (W0-05)

ND : Kira-kira apa saja yang diperlukan untuk mendukung proses penanganan insiden?

AG : Di level teknis perlu adanya tools keamanan yang tepat seperti SIEM dan manajemen log terpusat. Kemudian di level strategis perlu adanya regulasi, tata kelola, awareness dan compliance, dan standar keamanan aplikasi, terutama aplikasi website. (W0-06)

ND : Terkhir, apa harapan Bapak terhadap insiden ini?

AG : Tentunya ke depan insiden ini tidak terjadi lagi. Dan, seperti yang dibahas di awal, bahwa perlu adanya kebijakan strategis agar permasalahan terutama terkait pembuatan aplikasi tidak menjadi bom waktu bagi security di BPS. (W0-07)

**Lampiran 2** Transkrip Wawancara dengan Staf Jaringan Komunikasi Data

Nama Narasumber: Wildan dan Wahid

Jabatan: Staf Subdirektorat Jaringan Komunikasi Data (Tim Teknis Penanganan Insiden Intrusi)

Hari/Tanggal: Senin /25 Februari 2019

Kode: W1

ND: Nia Dwi Rahayuningtyas

W1: Wildan

W2: Wahid

ND : Seperti yang disebutkan dalam laporan, menurut teman-teman apakah yang menjadi penyebab utama terjadinya serangan website di BPS?

W1 : Menurut saya karena banyaknya *vulnerability* pada aplikasi website (W1-01). Terus kurang kooperatifnya pihak *developer* ketika diberitahu oleh pihak infra terkait adanya *vulnerability* tersebut (W1-02).

W2 :Iya seperti yang dibilang Wildan. Lebih ke kurangnya pemahaman dan komunikasi dari *developer* aplikasi (W1-03).

ND : Apakah sudah ada langkah yang dilakukan untuk menangani insiden tersebut ? Apa saja langkahnya?

W1 dan W2: sudah ada mba, kami menutup server untuk sementara kemudian memindahkan seluruh data penting ke server lain (W1-04).

ND : Apakah sudah ada kebijakan mengenai mekanisme/prosedur/langkah deteksi, prevensi, dan penanganan insiden serangan website ?

W1 : setahu kami belum ada kebijakan tertulis (W1-05).

W2 : kebijakan masih dalam bentuk kesepakatan dari tim teknis, belum secara tertulis (W1-06)

ND : Apa saja kendala yang dihadapi dalam penanganan insiden tersebut?

W1 : Tidak semua memahami masalah tersebut (W1-07). Dan pihak *developer* kurang paham tentang keamanan system (W1-08).

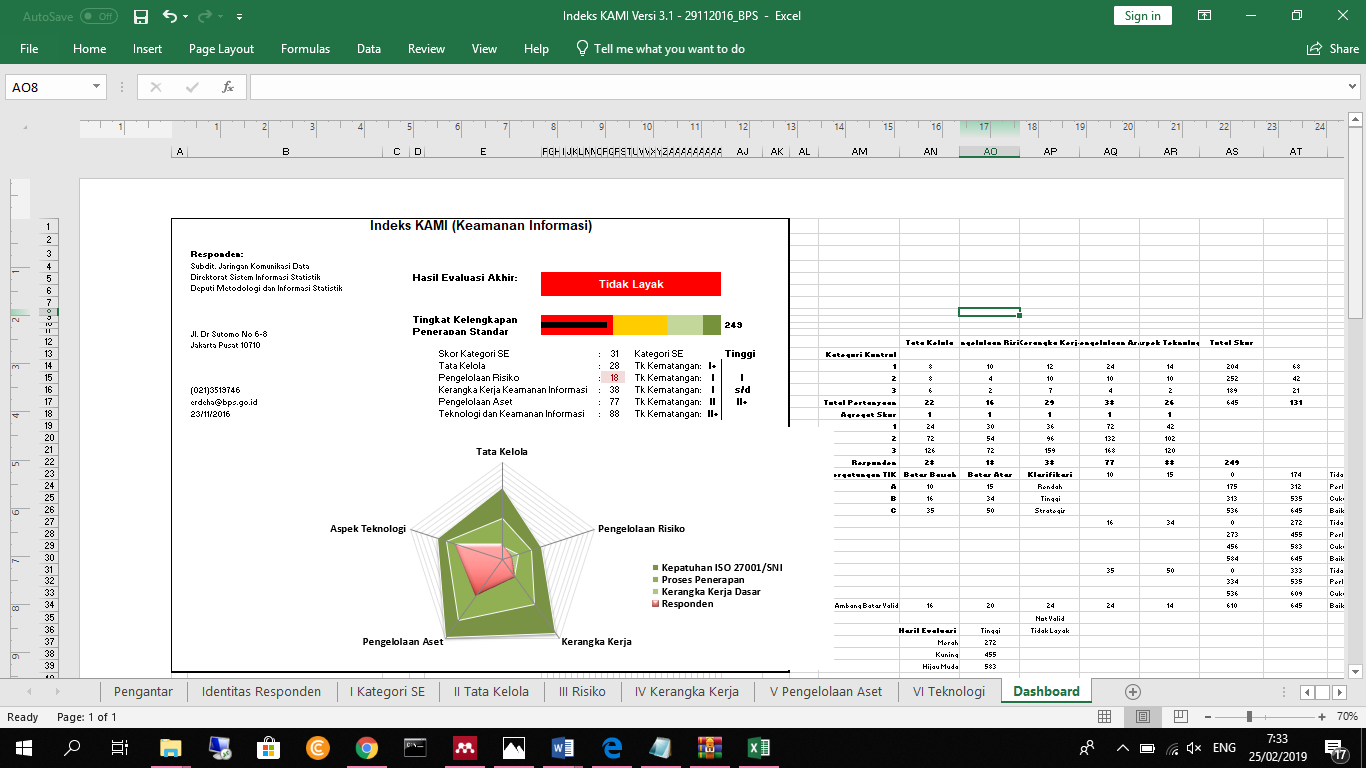
W2 : Iya kurangnya pemahaman dari *developer* aplikasi , dan kurangnya pengujian sebelum diluncurkannya aplikasi (W1-09).

ND : Kira-kira apa saja yang diperlukan untuk mendukung proses penanganan insiden?

W1 : banyak mba... tapi yang utama adalah komunikasi antara tim infra dengan pengembang system (W1-10).

W2 : kalau di kita lebih perlu di sisi teknis seperti IDS dan IPS mba (W1-11). Tapi terkait kebijakan atau kesepakatan tertulis antara tim teknis dan *developer* juga perlu (W1-12). Terus, pengujian aplikasi secara menyeluruh sebelum diluncurkannya aplikasi (W1-12). Dan juga perlu adanya komunikasi yang baik antara tim teknis dan *developer* (W1-13).

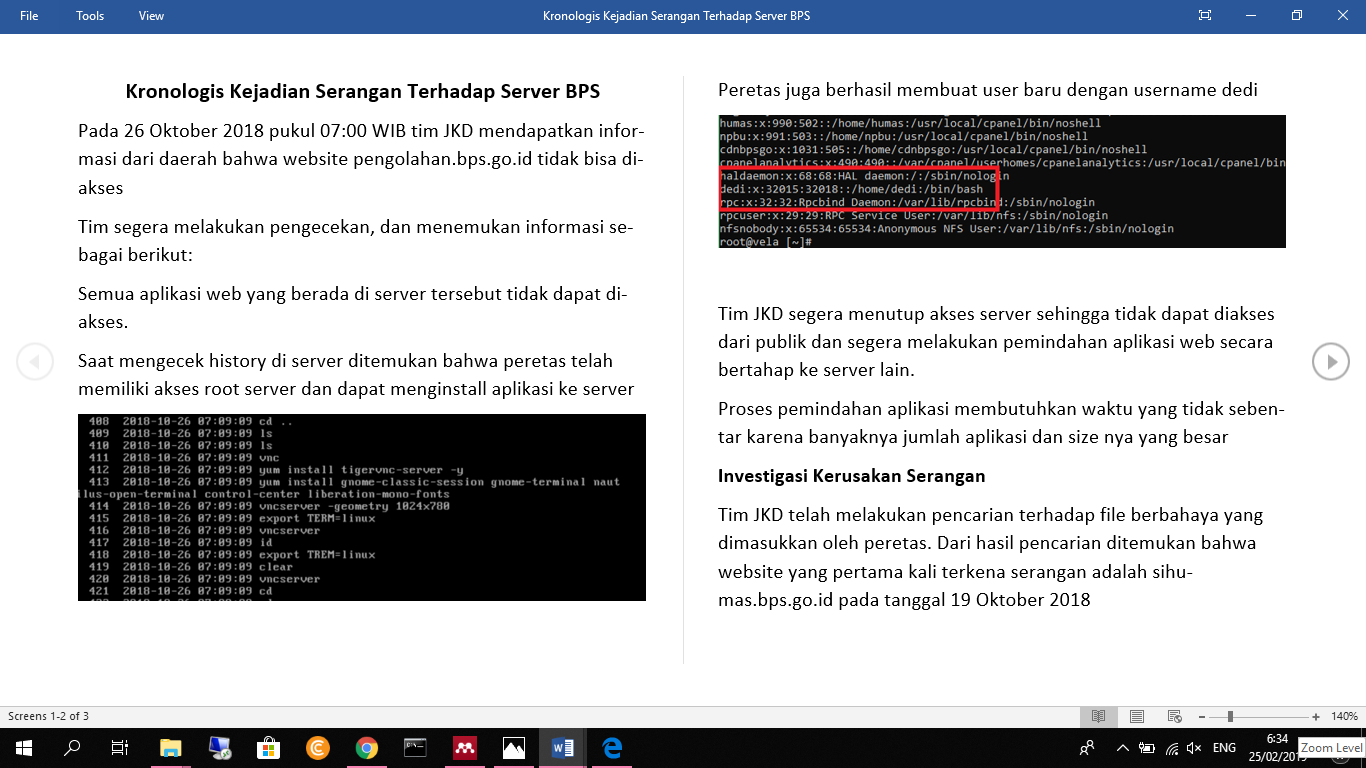
**Lampiran 3** Hasil Self Assessment BPS Menggunakan Index KAMI 3.1 Tahun 2016

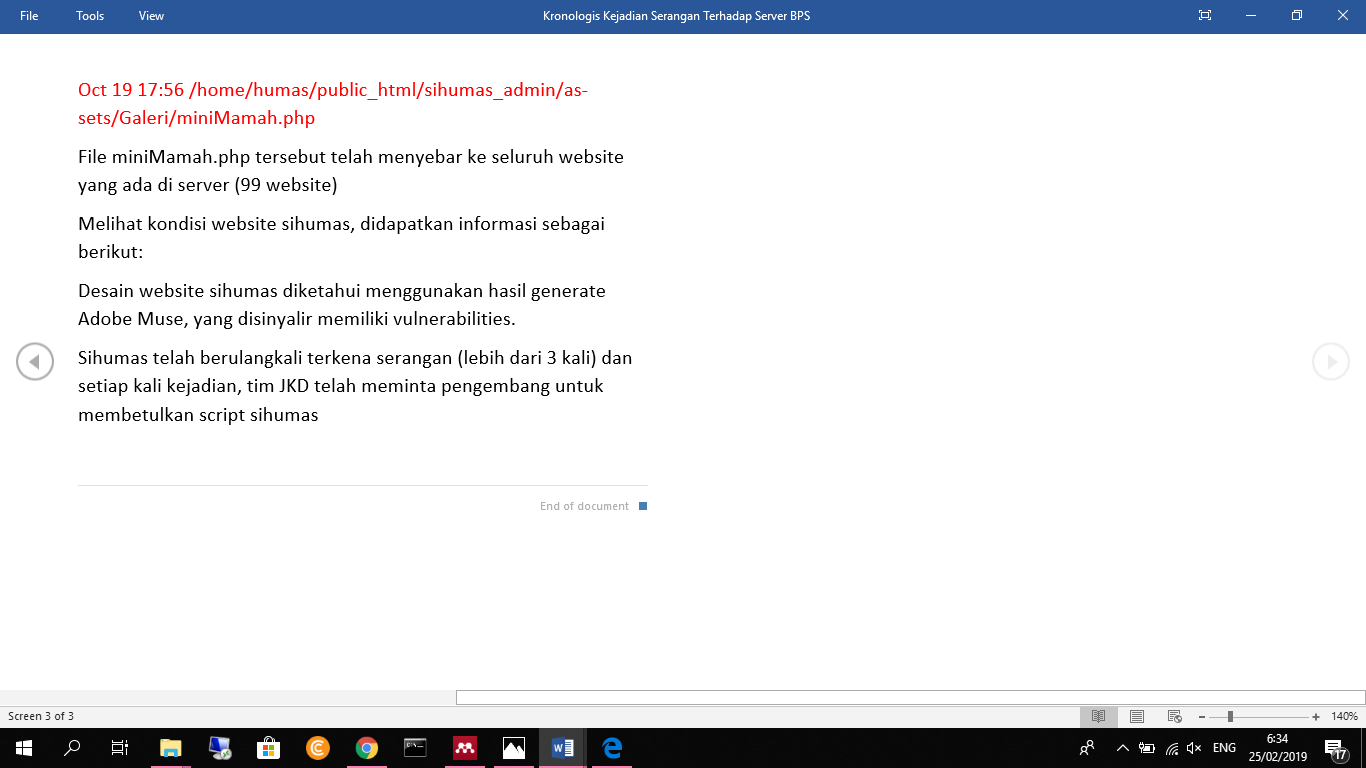


**Lampiran 4** Raw Log Insiden Intrusi 99 Website Oktober 2018

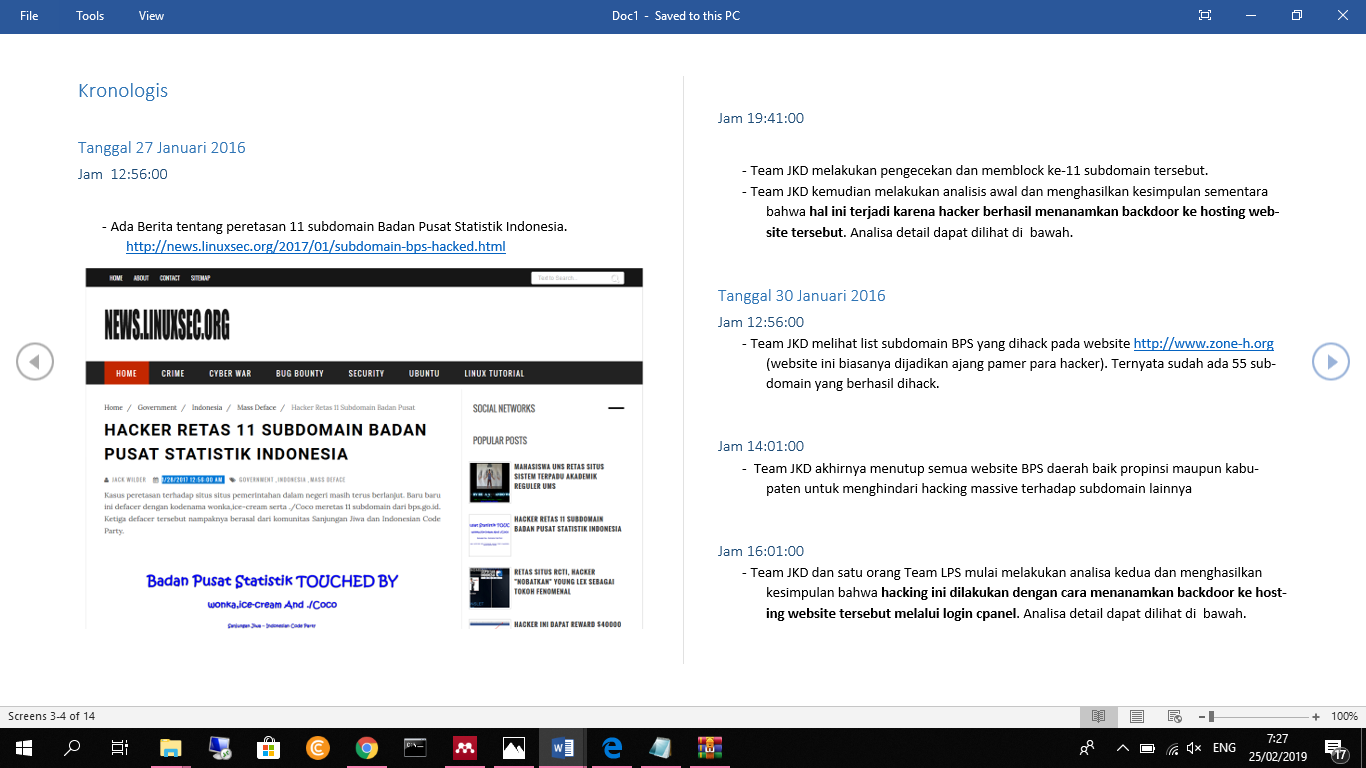
|  |
| --- |
| 60867297 12 -rw-r--r-- 1 skd skd 8722 Oct 25 21:12 /home/skd/public\_html/miniMamah.php  92012583 12 -rw-r--r-- 1 neracamln neracamln 8722 Oct 25 21:12 /home/neracamln/public\_html/miniMamah.php  12075403 12 -rw-r--r-- 1 root cpnszero 8722 Oct 25 21:12 /home/cmail/public\_html/miniMamah.php  411580 12 -rw-r--r-- 1 transportasibps transportasibps 8722 Oct 25 21:12 /home/transportasibps/public\_html/miniMamah.php  52741001 12 -rw-r--r-- 1 bps7400 bps7400 8722 Oct 25 21:12 /home/bps7400/public\_html/miniMamah.php  23904291 12 -rw-r--r-- 1 lotus lotus 8722 Oct 25 21:12 /home/lotus/public\_html/miniMamah.php  37625935 12 -rw-r--r-- 1 webpaguyubban webpaguyubban 8722 Oct 25 21:12 /home/webpaguyubban/public\_html/miniMamah.php  70959145 12 -rw-r--r-- 1 bps3600 bps3600 8722 Oct 25 21:12 /home/bps3600/public\_html/miniMamah.php  90349572 12 -rw-r--r-- 1 webbeta webbeta 8722 Oct 25 21:12 /home/webbeta/public\_html/miniMamah.php  74828207 12 -rw-r--r-- 1 jkdbpsgo jkdbpsgo 8722 Oct 25 21:12 /home/jkdbpsgo/public\_html/miniMamah.php  2686980 24 -rw-r--r-- 1 humas humas 22446 Oct 19 17:56 /home/humas/public\_html/sihumas\_admin/assets/Galeri/miniMamah.php  130131354 12 -rw-r--r-- 1 humas humas 8722 Oct 25 21:12 /home/humas/public\_html/miniMamah.php  11952473 12 -rw-r--r-- 1 surveiga surveiga 8722 Oct 25 21:12 /home/surveiga/public\_html/miniMamah.php  93495433 12 -rw-r--r-- 1 cdnbpsgo cdnbpsgo 8722 Oct 25 21:12 /home/cdnbpsgo/public\_html/miniMamah.php  36872516 12 -rw-r--r-- 1 bps7300 bps7300 8722 Oct 25 21:12 /home/bps7300/public\_html/miniMamah.php  41518731 12 -rw-r--r-- 1 statcap statcap 8722 Oct 25 21:12 /home/statcap/public\_html/miniMamah.php  36367223 12 -rw-r--r-- 1 bps7100 bps7100 8722 Oct 25 21:12 /home/bps7100/public\_html/miniMamah.php  36408057 12 -rw-r--r-- 1 loganalyticdev loganalyticdev 8722 Oct 25 21:12 /home/loganalyticdev/public\_html/miniMamah.php  82452482 12 -rw-r--r-- 1 ksabpsgo ksabpsgo 8722 Oct 25 21:12 /home/ksabpsgo/public\_html/miniMamah.php  44654806 12 -rw-r--r-- 1 moibsbps moibsbps 8722 Oct 25 21:12 /home/moibsbps/public\_html/miniMamah.php  18276353 12 -rw-r--r-- 1 bps6400 bps6400 8722 Oct 25 21:12 /home/bps6400/public\_html/miniMamah.php  61546548 12 -rw-r--r-- 1 reposito reposito 8722 Oct 25 21:12 /home/reposito/public\_html/miniMamah.php  93044777 12 -rw-r--r-- 1 bps5300 bps5300 8722 Oct 25 21:12 /home/bps5300/public\_html/miniMamah.php  102154243 12 -rw-r--r-- 1 bps1200 bps1200 8722 Oct 25 21:12 /home/bps1200/public\_html/miniMamah.php  121086123 12 -rw-r--r-- 1 monitoring monitoring 8722 Oct 25 21:12 /home/monitoring/public\_html/miniMamah.php  13402118 12 -rw-r--r-- 1 bps1800 bps1800 8722 Oct 25 21:12 /home/bps1800/public\_html/miniMamah.php  125239340 24 -rw-r--r-- 1 fms fms 22446 Oct 22 00:44 /home/fms/public\_html/menu/miniMamah.php  105136137 12 -rw-r--r-- 1 fms fms 8722 Oct 25 21:12 /home/fms/public\_html/miniMamah.php  87818262 12 -rw-r--r-- 1 tubps tubps 8722 Oct 25 21:12 /home/tubps/public\_html/miniMamah.php  63152225 12 -rw-r--r-- 1 pti pti 8722 Oct 25 21:12 /home/pti/public\_html/miniMamah.php  40454486 12 -rw-r--r-- 1 root cpnszero 8722 Oct 25 21:12 /home/jdih/public\_html/miniMamah.php  69017633 12 -rw-r--r-- 1 bps6200 bps6200 8722 Oct 25 21:12 /home/bps6200/public\_html/miniMamah.php  63187760 12 -rw-r--r-- 1 bps9100 bps9100 8722 Oct 25 21:12 /home/bps9100/public\_html/miniMamah.php  44007514 12 -rw-r--r-- 1 bps3300 bps3300 8722 Oct 25 21:12 /home/bps3300/public\_html/miniMamah.php  91031914 12 -rw-r--r-- 1 npbu npbu 8722 Oct 25 21:12 /home/npbu/public\_html/miniMamah.php  21872785 12 -rw-r--r-- 1 simojard simojard 8722 Oct 25 21:12 /home/simojard/public\_html/miniMamah.php  54534409 12 -rw-r--r-- 1 bps3500 bps3500 8722 Oct 25 21:12 /home/bps3500/public\_html/miniMamah.php  49250414 12 -rw-r--r-- 1 bps9400 bps9400 8722 Oct 25 21:12 /home/bps9400/public\_html/miniMamah.php  111476737 12 -rw-r--r-- 1 ratekbpsgo ratekbpsgo 8722 Oct 25 21:12 /home/ratekbpsgo/public\_html/miniMamah.php  1597982 12 -rw-r--r-- 1 spi spi 8722 Oct 25 21:12 /home/spi/public\_html/miniMamah.php  7938102 12 -rw-r--r-- 1 bendahara bendahara 8722 Oct 25 21:12 /home/bendahara/public\_html/miniMamah.php  53963131 12 -rw-r--r-- 1 silapor silapor 8722 Oct 25 21:12 /home/silapor/public\_html/miniMamah.php  54274965 12 -rw-r--r-- 1 surveiha surveiha 8722 Oct 25 21:12 /home/surveiha/public\_html/miniMamah.php  49897479 12 -rw-r--r-- 1 simpegbp simpegbp 8722 Oct 25 21:12 /home/simpegbp/public\_html/miniMamah.php  66283976 12 -rw-r--r-- 1 bps3100 bps3100 8722 Oct 25 21:12 /home/bps3100/public\_html/miniMamah.php  71658208 12 -rw-r--r-- 1 devrusabp devrusabp 8722 Oct 25 21:12 /home/devrusabp/public\_html/miniMamah.php  56566491 12 -rw-r--r-- 1 ukkbps ukkbps 8722 Oct 25 21:12 /home/ukkbps/public\_html/miniMamah.php  92299271 12 -rw-r--r-- 1 ppidbps ppidbps 8722 Oct 25 21:12 /home/ppidbps/public\_html/miniMamah.php  106938612 12 -rw-r--r-- 1 pbddev pbddev 8722 Oct 25 21:12 /home/pbddev/public\_html/miniMamah.php  3129347 12 -rw-r--r-- 1 bps7200 bps7200 8722 Oct 25 21:12 /home/bps7200/public\_html/miniMamah.php  37153701 12 -rw-r--r-- 1 deproduksi deproduksi 8722 Oct 25 21:12 /home/deproduksi/public\_html/miniMamah.php  64169250 12 -rw-r--r-- 1 jkd jkd 8722 Oct 25 21:12 /home/jkd/public\_html/miniMamah.php  17620993 12 -rw-r--r-- 1 sirusa sirusa 8722 Oct 25 21:12 /home/sirusa/public\_html/miniMamah.php  67985977 12 -rw-r--r-- 1 bps5200 bps5200 8722 Oct 25 21:12 /home/bps5200/public\_html/miniMamah.php  67545660 12 -rw-r--r-- 1 se2016 se2016 8722 Oct 25 21:12 /home/se2016/public\_html/miniMamah.php  2089767 12 -rw-r--r-- 1 shprbbpsgo shprbbpsgo 8722 Oct 25 21:12 /home/shprbbpsgo/public\_html/miniMamah.php  31015636 12 -rw-r--r-- 1 silastikbpsgo silastikbpsgo 8722 Oct 25 21:12 /home/silastikbpsgo/public\_html/miniMamah.php  81485840 12 -rw-r--r-- 1 jabalnusrabps jabalnusrabps 8722 Oct 25 21:12 /home/jabalnusrabps/public\_html/miniMamah.php  62193670 12 -rw-r--r-- 1 supportcpanel supportcpanel 8722 Oct 25 21:12 /home/supportcpanel/public\_html/miniMamah.php  85893125 12 -rw-r--r-- 1 pancpns pancpns 8722 Oct 25 21:12 /home/pancpns/public\_html/miniMamah.php  25894913 12 -rw-r--r-- 1 root root 8722 Oct 25 23:58 /home/bps1100/public\_html/aceh.bps.go.id/miniMamah.php  72089682 12 -rw-r--r-- 1 bps1100 bps1100 8722 Oct 25 21:12 /home/bps1100/public\_html/miniMamah.php  83312641 12 -rw-r--r-- 1 bps1100 bps1100 8722 Oct 26 00:00 /home/bps1100/public\_html/themes/miniMamah.php  11346197 12 -rw-r--r-- 1 knpeng knpeng 8722 Oct 25 21:12 /home/knpeng/public\_html/miniMamah.php  34899017 12 -rw-r--r-- 1 bps7500 bps7500 8722 Oct 25 21:12 /home/bps7500/public\_html/miniMamah.php  18416399 12 -rw-r--r-- 1 aduan aduan 8722 Oct 25 21:12 /home/aduan/public\_html/miniMamah.php  20856833 12 -rw-r--r-- 1 psks psks 8722 Oct 25 21:12 /home/psks/public\_html/miniMamah.php  100081914 12 -rw-r--r-- 1 bps5100 bps5100 8722 Oct 25 21:12 /home/bps5100/public\_html/miniMamah.php  125198361 12 -rw-r--r-- 1 bps6300 bps6300 8722 Oct 25 21:12 /home/bps6300/public\_html/miniMamah.php  81354763 12 -rw-r--r-- 1 ipmbpsgo ipmbpsgo 8722 Oct 25 21:12 /home/ipmbpsgo/public\_html/miniMamah.php  54816284 12 -rw-r--r-- 1 pemetaan pemetaan 8722 Oct 25 21:12 /home/pemetaan/public\_html/miniMamah.php  20160799 12 -rw-r--r-- 1 simojar simojar 8722 Oct 25 21:12 /home/simojar/public\_html/miniMamah.php  33702189 12 -rw-r--r-- 1 bps3200 bps3200 8722 Oct 25 21:12 /home/bps3200/public\_html/miniMamah.php  29859841 12 -rw-r--r-- 1 bps6500 bps6500 8722 Oct 25 21:12 /home/bps6500/public\_html/miniMamah.php  74399745 12 -rw-r--r-- 1 bps1700 bps1700 8722 Oct 25 21:12 /home/bps1700/public\_html/miniMamah.php  43459138 12 -rw-r--r-- 1 sp2010 sp2010 8722 Oct 25 21:12 /home/sp2010/public\_html/miniMamah.php  65757228 12 -rw-r--r-- 1 shk shk 8722 Oct 25 21:12 /home/shk/public\_html/miniMamah.php  114991122 12 -rw-r--r-- 1 bps7600 bps7600 8722 Oct 25 21:12 /home/bps7600/public\_html/miniMamah.php  115777558 12 -rw-r--r-- 1 bps1600 bps1600 8722 Oct 25 21:12 /home/bps1600/public\_html/miniMamah.php  60358754 12 -rw-r--r-- 1 ipd ipd 8722 Oct 25 21:12 /home/ipd/public\_html/miniMamah.php  126566430 12 -rw-r--r-- 1 ibrbps ibrbps 8722 Oct 25 21:12 /home/ibrbps/public\_html/miniMamah.php  95356813 12 -rw-r--r-- 1 pae pae 8722 Oct 25 21:12 /home/pae/public\_html/miniMamah.php  30171137 12 -rw-r--r-- 1 bps6100 bps6100 8722 Oct 25 21:12 /home/bps6100/public\_html/miniMamah.php  121782296 12 -rw-r--r-- 1 rbbps rbbps 8722 Oct 25 21:12 /home/rbbps/public\_html/miniMamah.php  61698074 12 -rw-r--r-- 1 bps8200 bps8200 8722 Oct 25 21:12 /home/bps8200/public\_html/miniMamah.php  66216651 12 -rw-r--r-- 1 akuntan akuntan 8722 Oct 25 21:12 /home/akuntan/public\_html/miniMamah.php  87007233 12 -rw-r--r-- 1 root cpnszero 8722 Oct 25 21:12 /home/lapkerbp/public\_html/miniMamah.php  99887476 12 -rw-r--r-- 1 jafung jafung 8722 Oct 25 21:12 /home/jafung/public\_html/miniMamah.php  63276151 12 -rw-r--r-- 1 bps8100 bps8100 8722 Oct 25 21:12 /home/bps8100/public\_html/miniMamah.php  52907223 12 -rw-r--r-- 1 bps3400 bps3400 8722 Oct 25 21:12 /home/bps3400/public\_html/miniMamah.php  63963222 12 -rw-r--r-- 1 bps1300 bps1300 8722 Oct 25 21:12 /home/bps1300/public\_html/miniMamah.php  6924550 12 -rw-r--r-- 1 kspm kspm 8722 Oct 25 21:12 /home/kspm/public\_html/miniMamah.php  74033540 12 -rw-r--r-- 1 bps1900 bps1900 8722 Oct 25 21:12 /home/bps1900/public\_html/miniMamah.php  23232559 12 -rw-r--r-- 1 st2013de st2013de 8722 Oct 25 21:12 /home/st2013de/public\_html/miniMamah.php  18211221 12 -rw-r--r-- 1 binagram binagram 8722 Oct 25 21:12 /home/binagram/public\_html/miniMamah.php  106381386 12 -rw-r--r-- 1 pustaka pustaka 8722 Oct 25 21:12 /home/pustaka/public\_html/miniMamah.php  61991263 12 -rw-r--r-- 1 bps1500 bps1500 8722 Oct 25 21:12 /home/bps1500/public\_html/miniMamah.php  92291076 12 -rw-r--r-- 1 bps2100 bps2100 8722 Oct 25 21:12 /home/bps2100/public\_html/miniMamah.php  29 12 -rw-r--r-- 1 bps9100 bps9100 8722 Oct 25 21:12 /home/miniMamah.php  103236188 12 -rw-r--r-- 1 statkerawanan statkerawanan 8722 Oct 25 21:12 /home/statkerawanan/public\_html/miniMamah.php  29974834 12 -rw-r--r-- 1 statpe statpe 8722 Oct 25 21:12 /home/statpe/public\_html/miniMamah.php  120979568 12 -rw-r--r-- 1 rekrutmenbpsgo rekrutmenbpsgo 8722 Oct 25 21:12 /home/rekrutmenbpsgo/public\_html/miniMamah.php  37153700 12 -rw-r--r-- 1 cpnszero cpnszero 8722 Oct 25 21:12 /home/cpnszero/public\_html/miniMamah.php  123396147 12 -rw-r--r-- 1 pis pis 8722 Oct 25 21:12 /home/pis/public\_html/miniMamah.php  60444253 12 -rw-r--r-- 1 sirusabackend sirusabackend 8722 Oct 25 21:12 /home/sirusabackend/public\_html/miniMamah.php  120750082 12 -rw-r--r-- 1 bps1400 bps1400 8722 Oct 25 21:12 /home/bps1400/public\_html/miniMamah.php |

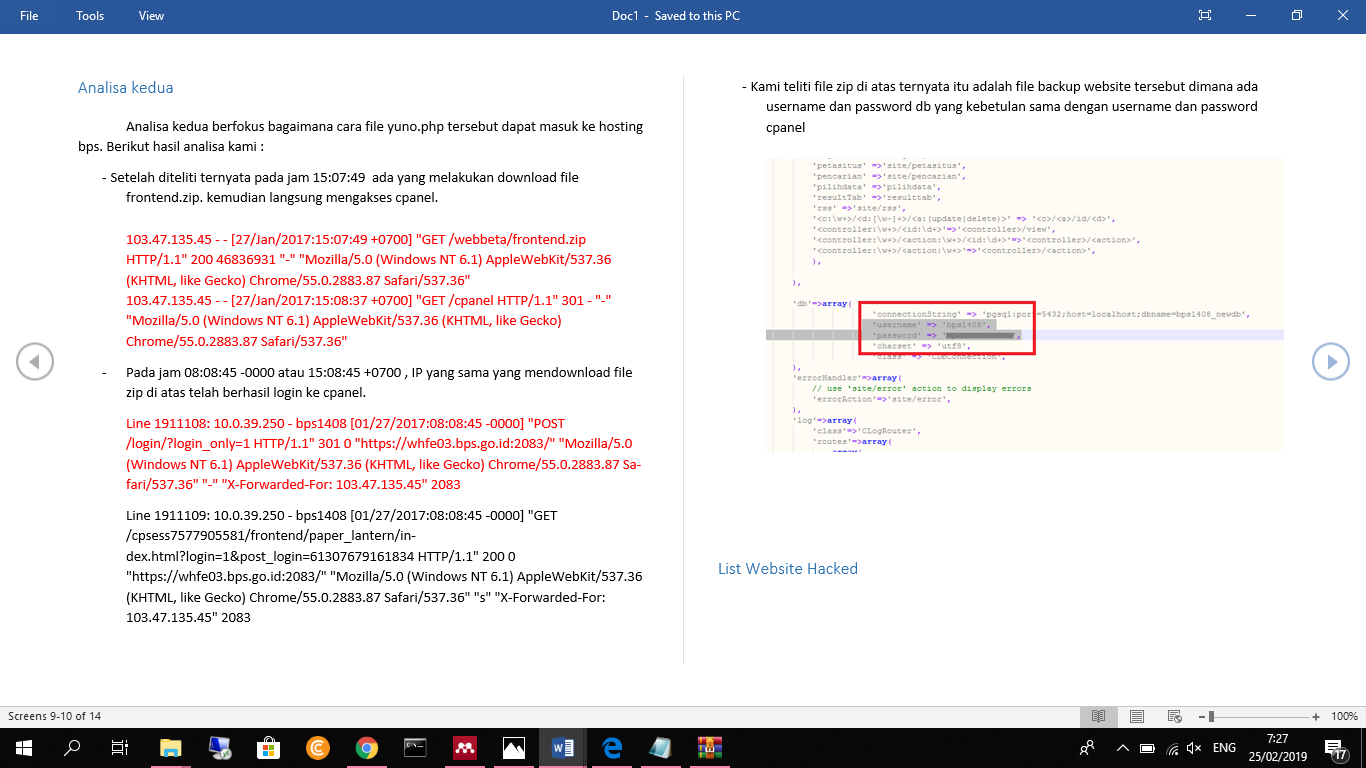
**Lampiran 5** Laporan Kronologi Insiden Intrusi Website Oktober 2018

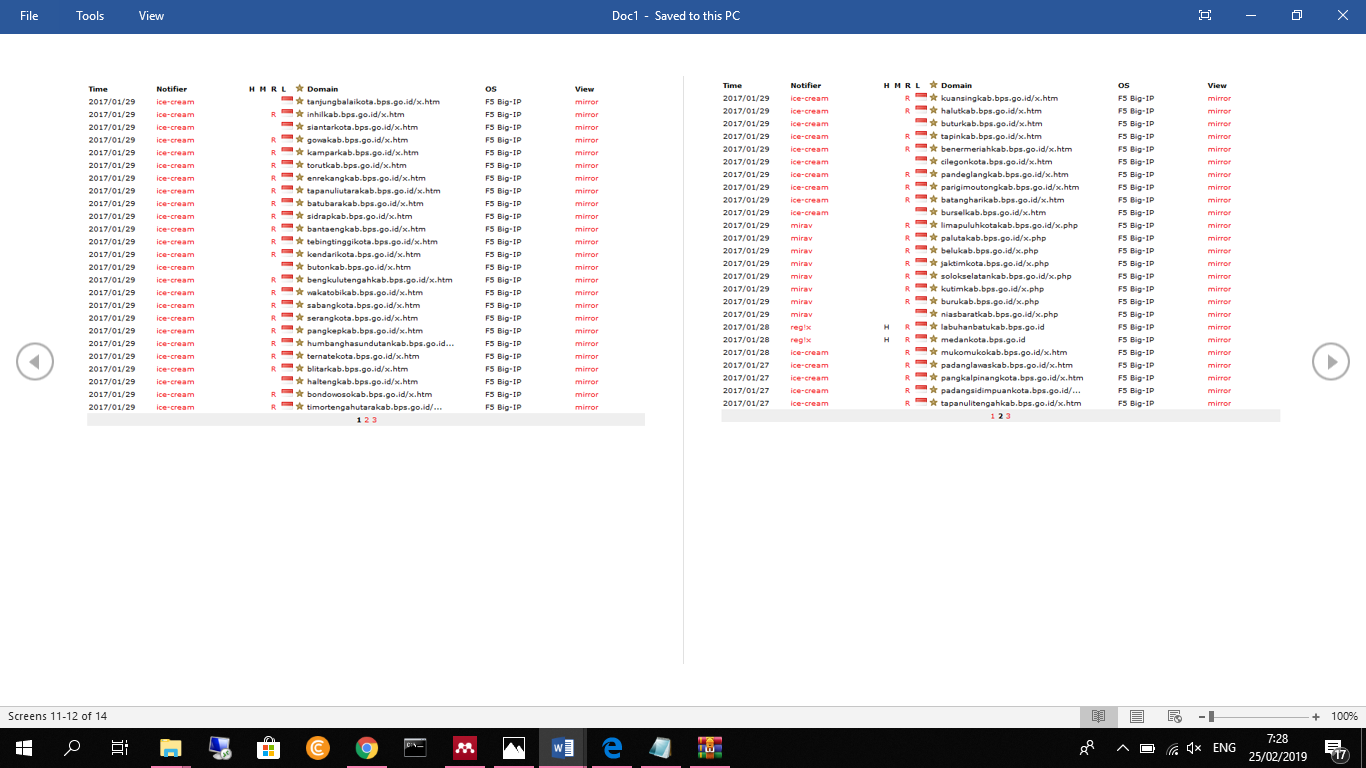




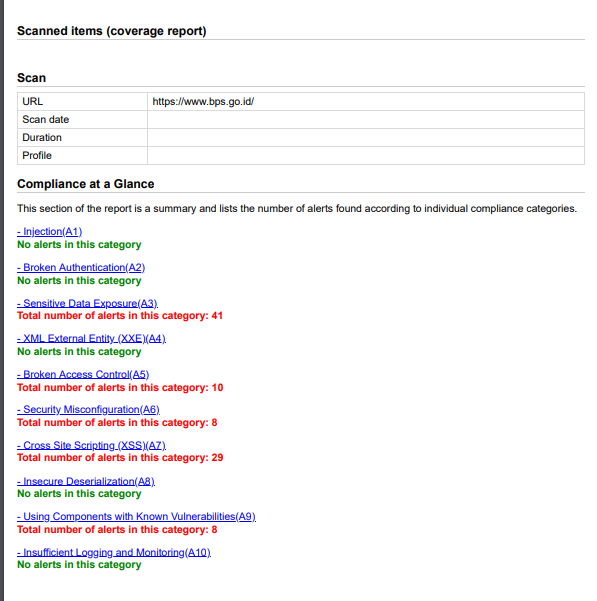
**Lampiran 6** Laporan Kronologi Insiden Intrusi Website Januari 2017



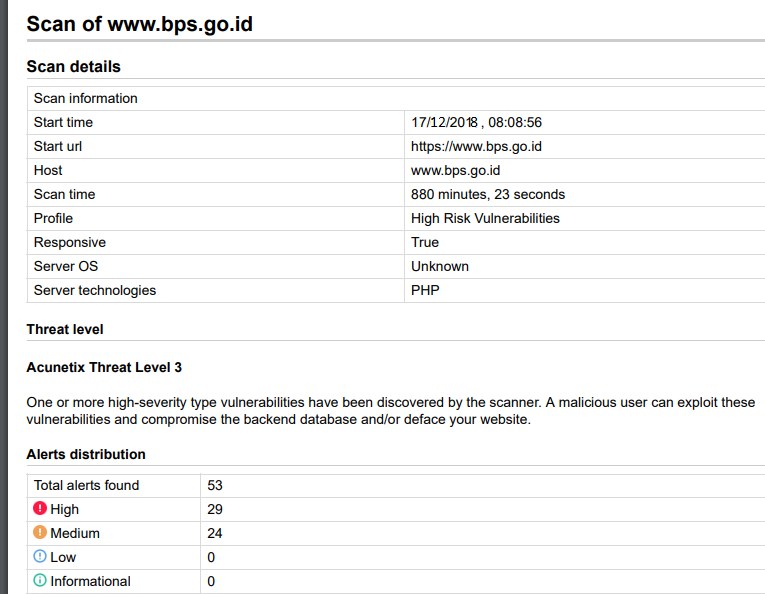




**Lampiran 7** Laporan Hasil Vulnerability Assessment Web .bps.go.id Desember 2018



OKGDS2Dd2



**Lampiran 8** Transkrip Wawancara dengan Kepala Seksi Pengembangan Jaringan Komunikasi Data

Nama Narasumber: Rosita Dewi Hadiyanti

Jabatan: Kepala Seksi Pengembangan Jaringan Komunikasi Data

Hari/Tanggal: Selasa /19 Maret 2019

Kode: W

N: Nia Dwi Rahayuningtyas

N : Seberapa penting keamanan website BPS? Apakah ada rujukannya/pernyataan tertulis?

W : Penting, tapi memang tidak tertulis khusus untuk keamanan aplikasi web, Kalau melihat dari indikator kinerja adanya lebih bersifat keamanan secara umum. Layanan JKD (Jaringan komunikasi data) punya SLA akan tetapi ketentuannya masih belum official.

N : Apakah pernah ada laporan kasus kehilangan data, pengubahan konten, dan terhentinya layanan?

W : Pernah, ada web monitoring survei dan sensus dimana web mengalami content defacement, data mengalami penurunan progres, dan data terakhir hilang dari database (W3-01). Penyelesaiannya dengan melakukan restore data terakhir dan menutup akses website dari luar, hanya bisa dari VPN. Kalau kasus yang lama, pernah ada pencurian data terkait PNBP (Penerimaan Negara Bukan Pajak) dimana pihak yang tidak bertanggung jawab menjual data tersebut melalui sebuah web (W3-02). Akan tetapi kasus ini sudah selesai.

N: Dampak kerugian yang pernah ditimbulkan dari pelanggaran keamanan/*security breach* apa?

W : Lebih ke reputasi sih. (W3-03)

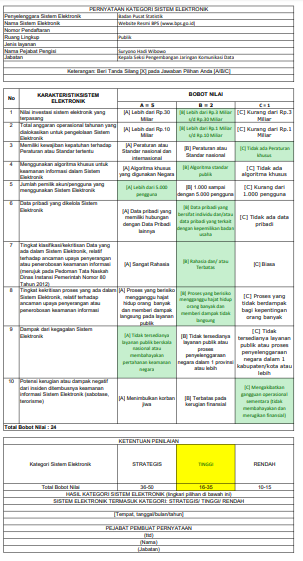
N : Apakah pernah ada komplain dari pengguna ketika terjadi insiden?

W : Pernah, lebih sering melalui telepon. (W3-04)

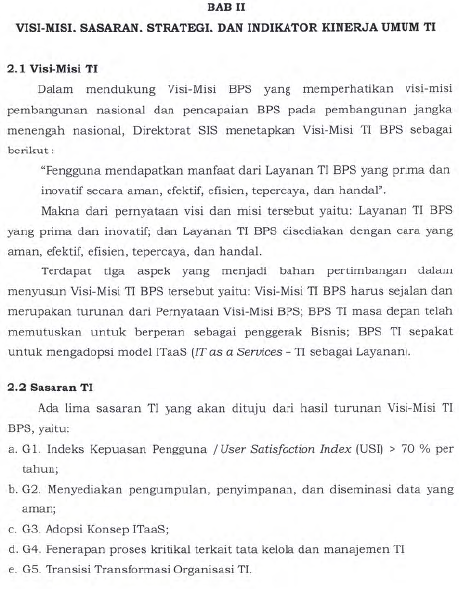
N : Rata-rata lama penanganan insiden berapa lama?

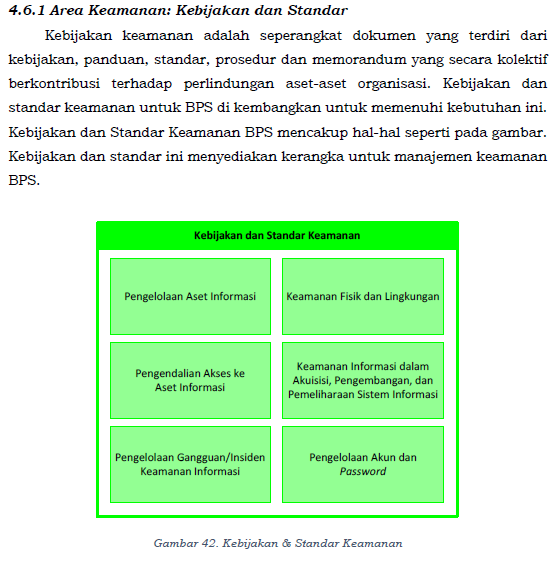
W : Tergantung keparahannya, kalau yang kasus oktober 2018 kemarin dibutuhkan waktu dengan interval 1-3 hari (W3-05) untuk beberapa web yang penting, akan tetapi sampai saat ini masih ada web yang belum dilakukan perbaikan oleh developernya, jadi masih ditutup.

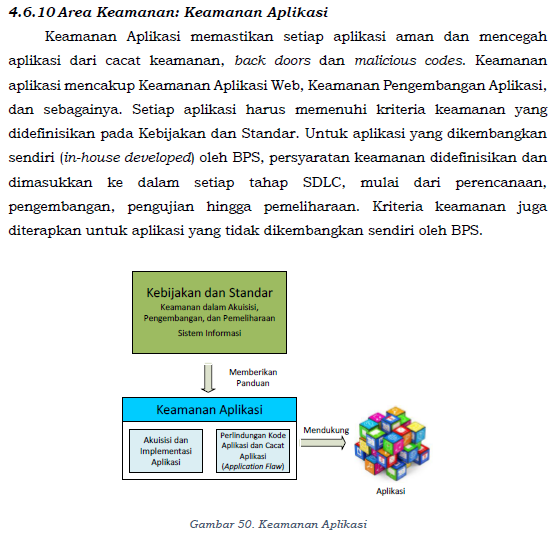
**Lampiran 9** Pernyataan Kategori Sistem Elektronik (Snapshoot [www.bps.go.id](http://www.bps.go.id))



**Lampiran 10** Visi-Misi dan Sasaran TI BPS (Perka BPS No 94 Tahun 2018)







**Lampiran 11** Contoh Lembar Observasi Developer

**LEMBAR OBSERVASI DEVELOPER**

**(OWASP ASVS 4.0)**

|  |
| --- |
|  |

Application :

Developer :

Work Unit :

|  |
| --- |
| Fill in the check list for the 14 security controls that develop the following applications according to the level of security implementation.  Note:  L1: Not yet implementing security  L2: Applying partially  L3: Apply fully |

|  |  |  |  |
| --- | --- | --- | --- |
| **V1.1 Secure Software Development Lifecycle Requirements** | **L1** | **L2** | **L3** |
| **1.1.1** Verify the use of a secure software development lifecycle that addresses security in all stages of development. |  |  |  |
| **1.1.2** Verify the use of threat modeling for every design change or sprint planning to identify threats, plan for countermeasures, facilitate appropriate risk responses, and guide security testing. |  |  |  |
| **1.1.3** Verify that all user stories and features contain functional security constraints, such as "As a user, I should be able to view and edit my profile. I should not be able to view or edit anyone else's profile" |  |  |  |
| **1.1.4** Verify documentation and justification of all the application's trust boundaries, components, and significant data flows. |  |  |  |
| **1.1.5** Verify definition and security analysis of the application's high-level architecture and all connected remote services. |  |  |  |
| **1.1.6** Verify implementation of centralized, simple (economy of design), vetted, secure, and reusable security controls to avoid duplicate, missing, ineffective, or insecure controls. |  |  |  |
| **1.1.7** Verify availability of a secure coding checklist, security requirements, guideline, or policy to all developers and testers. |  |  |  |
| **V1.2 Authentication Architectural Requirements** | **L1** | **L2** | **L3** |
| **1.2.1** Verify the use of unique or special low-privilege operating system accounts for all application components, services, and servers. |  |  |  |
| **1.2.2** Verify that communications between application components, including APIs, middleware and data layers, are authenticated. Components should have the least necessary privileges needed. |  |  |  |
| **1.2.3** Verify that the application uses a single vetted authentication mechanism that is known to be secure, can be extended to include strong authentication, and has sufficient logging and monitoring to detect account abuse or breaches. |  |  |  |
| **1.2.4** Verify that all authentication pathways and identity management APIs implement consistent authentication security control strength, such that there are no weaker alternatives per the risk of the application. |  |  |  |