SISTEM DETEKSI CELAH KEAMANAN PADA APLIKASI WEB DI UNIVERSITAS BRAWIJAYA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
ABDULLAH
NIM: 125150207111002

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

SISTEM DETEKSI CELAH KEAMANAN PADA APLIKASI WEB DI UNIVERSITAS BRAWIJAYA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
ABDULLAH
NIM: 125150207111002

Skripsi ini telah diuji dan dinyatakan lulus pada
11 Agustus 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Eko Sakti P.,S.Kom, M.Kom	Kasyful Amron, S.T, M.Sc
NIK: 201102 860805 1 001	NIP: 19750803 200312 1 003

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Agustus 2016

Abdullah

NIM: 125150207111002

KATA PENGANTAR

Puji Syukur penulis panjarkan kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Sistem Deteksi Celah Keamanan Pada Aplikasi Web di Universitas Brawijaya" dengan baik. Penyusunan skripsi ini dapat terlaksana dengan baik karena adanya bantuan secara langsung maupun tidak langsung dari pihak tertentu, diantaranya:

1. Bapak Eko Sakti P., S.Kom. selaku dosen pembimbing I yang telah memberikan ilmu dan saran untuk skripsi ini
2. Bapak Kasyful Amron, S.T, M.Sc. selaku dosen pembimbing II yang telah memberikan ilmu dan saran untuk skripsi ini
3. Orang tua, yang telah memberikan dukungan moral dan material
4. Komunitas Indonesian Coder dan Malang Cyber Crew yang telah memberikan banyak ilmu dan membantu dalam proses penyelesaian skripsi ini
5. Teman-teman yang telah membantu memberi saran dan kritik atas skripsi ini

Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan jauh dari sempurna, karena keterbatasn materi dan pengetahuan yang dimiliki penulis. Akhirnya semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca terutama mahasiswa Informatika Universitas Brawijaya

Malang, 12 Agustus 2016

Penulis

abdullahtracker@gmail.com

ABSTRAK

Abdullah. 2016. Sistem Deteksi Celah Keamanan Pada Aplikasi Web di Universitas Brawijaya. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Eko Sakti P., S.Kom, M.Kom dan Kasyful Amron, S.T, M.Sc.

Universitas Brawijaya memiliki website dengan berbagai domain dan subdomain, masing-masing website dikelola oleh administrator yang berbeda pada masing-masing unit. Adanya perbedaan administrasi pada website menimbulkan permasalahan seperti kurangnya pengetahuan administrator tentang celah keamanan dan tidak adanya informasi tentang celah keamanan dari penyedia *hosting* sehingga beberapa administrator kurang tanggap jika terjadi kasus *hacking* pada website yang dikelola.Untuk mengatasi hal tersebut dikembangkan sebuah sistem yang dapat mendeteksi celah keamanan pada website yang berada di Universitas Brawijaya.Sistem yang dikembangkan memiliki kemampuan dalam mendeteksi celah keamanan berdasarkan jenis website.Sistem ini dapat digunakan oleh pengelola website di Universitas Brawijaya dalam mendeteksi celah keamanan sebelum sebuah website dipublikasikan.Berdasarkan hasil pengujian,didapatkan hasil bahwa sistem yang dikembangkan lebih efektif dalam mendeteksi celah keamanan pada website yang menggunakan WordPress. Teknik deteksi celah keamanan berdasarkan jenis website cukup efektif namun memiliki keterbatasan informasi berupa database celah keamanan pada jenis website tertentu.

Kata kunci: Keamanan website, Privasi data, Celah keamanan, *Hacking*.

ABSTRACT

Abdullah. 2016. Vulnerability Detection System on Web Application at Brawijaya University. Faculty of Computer Science, Brawijaya University, Malang. Lecturer: Eko Sakti P., S.Kom, M.Kom and Kasyful Amron, S.T, M.Sc.

Brawijaya University has many websites with different domains and subdomains, each website is managed by different administrators in each unit. The big difference in the administration of the website causing problems such as a lack of knowledge administrators about vulnerabilities and the absence of information about the vulnerability from the hosting provider so that some administrators are less responsive in case of hacking on a website they managed. To overcome these problems developed a system that can detect vulnerabilities in websites that are in the Brawijaya University. The system developed has the ability to detect security holes based on the type of website. This system can be used by the website manager at Brawijaya University in detecting security holes before a website published. Based on test results, showed that the developed system is more effective in detecting security holes in websites using WordPress. Vulnerability detection technique based on the type of website is quite effective but has limited information such as database security holes in certain types of websites.

Keywords: Website security, data privacy, Vulnerabilities, Hacking.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar belakang

Perkembangan teknologi mempengaruhi cara manusia dalam mengakses dan menyebarkan informasi. Di era modern ini, akses informasi dan penyebarannya dapat dilakukan melalui internet misalnya dengan menggunakan website. Website saat ini tidak hanya digunakan untuk menampilkan halaman informasi saja, namun juga banyak kebutuhan lainnya seperti keperluan bisnis, aplikasi internal perusahaan, pemerintahan dan bidang lainnya Dengan adanya website,membuat pekerjaan seseorang menjadi lebih mudah,misalnya kita tidak perlu membuka toko untuk berjualan karena telah tersedia website jual beli (e-commerce).Namun penggunaan website saat ini tidak hanya memiliki dampak positif tapi juga memiliki dampak negatif.Banyak kasus-kasus seperti pencurian data secara online, menyebarluaskan data-data penting dari sebuah perusaahan (*leaked data*), pemanfaatan sistem untuk kepentingan pribadi dan kasus-kasus lainnya yang berhubungan dengan *cybercrime*. Kasus *leaked data* pernah terjadi pada sebuah perusahaan pemasok software spionasi dari Italia yang bernama *Hacking Team*, peretas *Hacking Team* mengunduh sekitar 400GB database dan menyebarkannya di Internet, data-data yang disebarkan merupakan data yang penting seperti *source code* dari program spionase yang hanya diperuntukkan bagi lembaga pemerintahan, bahkan lembaga di Indonesia sempat menjadi sasaran klien oleh Hacking Team (Amalia, E. I,2015), hal ini diketahui setelah semua email dari Hacking Team diretas dan dapat diakses via WikiLeaks (Tarigan, I. A,2015).

Dalam Negara Indonesia, terdapat ratusan jenis website mulai dari pendidikan, pemerintahan, instansi dan website lainnya yang hampir setiap tahun diretas oleh *blackhat hacker*, hal ini bisa dilihat pada website Zone-H yang menyimpan database website-website hasil peretasan (Zone-H, 2015) sehingga tidak menutup kemungkinan para *hacker* ini telah melakukan eksploitasi pada sistem dan mencuri data-data penting dari lembaga-lembaga pemerintahan yang ada di Indonesia. Website-website perguruan tinggi juga sering menjadi sasaran bagi *blackhat hacker*. Menurut data dari Zone-H, hingga tahun 2015 terdapat setidaknya 11.369 kasus deface yang terjadi pada website dengan domain ac.id (Zone-H, 2015) termasuk website-website yang berada di Universitas Brawijaya.

Universitas Brawijaya memiliki banyak website yang terdiri dari berbagai macam subdomain. Masing-masing subdomain dikelola oleh administrator yang berbeda-beda di setiap unit (UBHosting,2014). Perbedaan pengelolaan website pada masing-masing unit dan kurangnya pengetahuan tentang celah keamanan bagi administrator menjadi penyebab banyaknya celah keamanan pada website. Menurut data dari Zone-H, setidaknya terdapat 103 kasus deface yang terjadi pada domain dan subdomain ub.ac.id (Zone-H, 2015).Kasus yang pernah terjadi adalah aksi *deface* pada website Eksekutif Mahasiswa Universitas Brawijaya dimana website ini menyimpan 13.000 data mahasiswa baru angkatan 2015. Data yang

disimpan berupa username,email, password dan data lainnya yang dapat diambil oleh peretas yang masuk ke dalam website tersebut.

Melihat permasalahan diatas, maka diusulkan sebuah sistem yang berfungsi untuk mendeteksi celah keamanan pada website di lingkungan Universitas Brawijaya yang menggunakan domain ub.ac.id,selain itu sistem ini juga akan memberikan laporan hasil deteksi yang selanjutnya akan dilaporkan ke pihak yang bertanggung jawab dalam mengelola website di Universitas Brawijaya.

1.2 Rumusan masalah

Berdasarkan paparan latar belakang tersebut, maka rumusan masalah yang dapat dikaji dalam penelitian ini sebagai berikut:

1. Bagaimana pendeteksian celah keamanan pada website-website di Universitas Brawijaya?
2. Bagaimana membangun sistem untuk mendeteksi celah kemanan pada aplikasi di web di Universitas Brawijaya?
3. Bagaimanan performa sistem yang telah dibangun dalam mendeteksi celah keamanan pada website?

1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian di skripsi ini yaitu:

1. Melakukan penelitian mengenai teknik yang tepat dalam mendeteksi celah keamanan pada website.
2. Mengetahui hasil pengujian celah keamanan pada website-website di Universitas Brawijaya
3. Dapat mengetahui hasil performa sistem yang telah dibangun

1.4 Manfaat

Manfaat yang diperoleh dari penelitian yang sudah dilakukan yaitu :

1. Bagi penulis

- Dapat mengimplementasikan ilmu yang didapat pada website pribadi.
- Menambah pengetahuan tentang keamanan pada website yang terdapat di Universitas Brawijaya

2. Bagi Pembaca

- Mendapatkan pengetahuan tentang pentingnya keamanan pada website
- Mendapatkan wawasan tentang sistem yang dapat mendeteksi kemananan pada website.

3. Untuk Universitas Brawijaya

- Sebagai bahan pembelajaran bagi mahasiswa yang ingin melakukan penelitian sejenis

- Mendapatkan tambahan referensi untuk meningkatkan keamanan pada website.
- Sebagai bahan referensi agar penelitian ini dapat juga digunakan oleh mahasiswa yang ingin melanjutkan riset atau penelitian selanjutnya.

1.5 Batasan Masalah

Sesuai dengan terpaparnya permasalahan yang terdapat pada latar belakang agar tidak memperluas area bahasan maka diperlukan suatu batasan sebagai berikut:

1. Pengujian ini bersifat simulasi.
2. Pengujian hanya dilakukan pada website yang menggunakan CMS (Content Management System) dan website PHP manual / framework.
3. Sistem deteksi dalam penelitian ini diimplementasikan pada sistem operasi Linux.

1.6 Sistematika Pembahasan

a) BAB I PENDAHULUAN

Menjelaskan tentang latar belakang, tujuan, rumusan masalah,manfaat penelitian, batasan masalah, dan sistematika penulisan.

b) BAB II TINJAUAN PUSTAKA

Menguraikan kajian pustakan dan dasar teori yang berkaitan dengan *penetration testing*, daftar celah keamanan yang sering terjadi dan statistik celah keamanan dunia.

c) BAB III METODOLOGI PENELITIAN

Membahas metodologi yang akan digunakan untuk menyelesaikan masalah-masalah dalam penelitian dan juga akan menjelaskan mengapa metode dan teknik tersebut dipilih dalam menyelesaikan masalah.

d) BAB IV DATA DAN LINGKUNGAN PENELITIAN

Membahas data-data yang digunakan untuk mendukung proses penelitian. Data-data ini digunakan sebagai acuan dalam pembuatan sistem deteksi celah keamanan serta dalam menunjang pembuatan data uji.

e) BAB IV PERANCARANGAN DAN IMPLEMENTASI

Membahas bagaimana merancang sistem dan melakukan implementasi pada sistem yang dikembangkan sesuai dengan kebutuhan fungsional yang telah didefinisikan dan selanjutnya dari hasil perancangan.

f) BAB V PENGUJIAN DAN ANALISIS

Membahas pengujian yang akan dilakukan pada sistem yang telah dikembangkan. Terdapat beberapa metode pengujian, yaitu mengukur performa dari sistem apakah dapat berjalan sesuai hasil yang diharapkan, dilakukan pengujian apakah sistem dapat melakukan deteksi sesuai dengan hasil perancangan dan selanjutnya akan dilakukan analisis terhadap hasil pengujian yang didapatkan.

g) BAB VI KESIMPULAN

Bab ini membahas tentang kesimpulan yang dapat diambil berdasarkan rumusan masalah yang telah ditetapkan sebelumnya. Serta memberikan saran terkait hal-hal yang perlu dikembangkan dalam penelitian ini agar penelitian ini dapat menjadi lebih baik lagi kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada pembahasan ini terdapat kajian pustaka yang merupakan peninjauan kembali pustaka – pustaka pendukung yang memiliki keterkaitan dengan penelitian ini. Dasar teori juga dibutuhkan sebagai teori dasar yang harus dipahami agar pembaca dapat mengerti terkait permasalahan yang dibahas dalam penelitian.

2.1 Penetration Testing

Penetration testing merupakan teknik yang digunakan untuk mendapatkan akses ke sistem organisasi atau melanggar sistem jaringan layaknya aktivitas illegal yang dilakukan oleh *blackhat hacker*. Namun dalam aktivitas penetration testing sedikit berbeda dari aktivitas *blackhat hacker*. Motivasi dalam penetration testing adalah untuk mengamankan sistem, bukan merusak atau mencuri data. Dalam melakukan *penetration testing* pada suatu organisasi, kita harus mendapatkan izin dari organisasi tersebut agar aktivitas yang kita lakukan tidak dianggap illegal (Kennedy, O'Gorman, & Aharoni, 2011).

Jika melihat banyak referensi, terdapat cara yang berbeda-beda dalam penetration testing. Akhirnya beberapa orang bingung dalam menentukan metode yang benar dalam penetration testing. Dari masalah ini muncul sebuah website *The Penetration Testing Execution Standard (PTES)* yang memberikan informasi tentang standar dalam penetration testing sehingga pada penelitian ini akan merujuk pada standar PTES.

PTES terbagi dalam 7 kategori, mencakup semua hal yang berhubungan dengan *penetration testing*.Dimulai dari pengumpulan informasi hingga proses eksploitasi pada sistem. 7 kategori tersebut adalah sebagai berikut:

a. Pre-engagement Interactions

   Pre-engagement Interactions merupakan tahapan pertama dalam *penetration testing*, pada tahapan ini seorang *penetration tester* melakukan diskusi dengan klien terkait ruang lingkup dan tujuan *penetration testing*. Pada tahapan ini, kita harus menjelaskan secara detail kegiatan yang dilakukan selama pentest mulai dari awal hingga membuat laporan pentest, hal ini dimaksudkan agar tidak terjadi salah faham antara klien dan *penetration tester*.

b. Intelligence Gathering

   Intelligence Gathering merupakan tahapan dimana kita mengumpulkan informasi-informasi penting dari target yang nantinya akan digunakan selama proses pentest contohnya melihat informasi di media social,Google Hacking, fingerprint,footprinting, web crawling dan lain-lain. Kemampuan yang harus dikuasai dalam tahap ini adalah memiliki kemampuan untuk belajar segala sesuatu tentang target, perilaku,sistem yang digunakan dan bagaimana sistem pengoperasiannya, dan akhirnya bagaimana sistem tersebut bisa diserang.

Dalam proses *intelligence gathering*, seorang penetration tester mencoba untuk mengidentifikasi mekanisme perlindungan apa yang terdapat pada target dengan perlahan-lahan dan berhati-hati dalam menyelidiki sistem tersebut. Sebagai contoh beberapa perusahaan mengakses suatu port pada sistem dan jika kita memaksa masuk ke port selain whitelist maka IP Address kita akan diblokir oleh sistem. Hal yang sama juga berlaku pada saat pentest ke sebuah web dimana pada batas waktu tertentu kita akan diblokir karena terdeteksi melakukan serangan. Hal-hal seperti harus bisa dipelajari dan dimengerti oleh seorang penetration tester agar proses pentest bisa dilakukan dengan baik.

c. Threat Modeling

Threat Modeling menggunakan informasi yang didapatkan dari *intelligence gathering* yang digunakan untuk mengidentifikasi kerentanan yang terdapat pada sistem. Pada tahapan *Threat Modeling,* kita akan menentukan metode serangan apa yang paling efektif.Hasil dari modeling ini nantinya akan menetntukan bagaimana sebuah sistem bisa diserang, dalam hal ini kita menganggap organisasi (klien) sebagai musuh untuk mencari kelemahan pada suatu sistem.

d. Vulnerability Analysis

Setelah mengidentifikasi metode serangan. Kita selanjutnya perlu mempertimbangkan bagaimana kita akan mengakses target untuk menjalankan serangan. Selama tahapan *Vulnerability Analysis* kita menggabungkan informasi-informasi yang didapatkan dari tahapan-tahapan sebelumnya dan memahaminya agar dapat mengetahui serangan apa yang layak digunakan. Contohnya informasi tentang port dan melakukan *vulnerability scanning* pada port tersebut, hasil dari *banner grabbing, OS fingerprint* dan lain-lain

e. Exploitation

Explotation merupakan tahap yang menentukan apakah sebuah sistem bisa diserang. Tahapan ini sering dilakukan dengan serangan yang dapat mengganggu sistem. Karena dalam proses exploitation, kita harus mengerti sistem apa seharusnya yang diserang, kita harus tahu bahwa sistem tersebut memiliki kerentan, jika kita melakukan explotation tanpa mengetahui kelemahannya, maka usaha kita tidak akan produktif dan bisa jadi mengganggu aktivitas sistem. Maka dari itu proses-proses dari tahapan sebelumnya sangat dibutuhkan.

f. Post Exploitation

Setelah melakukan exploitation,hal yang selanjutnya kita lakukan adalah *Post Exploitation*. Tahapan ini juga merupakan yang terpenting. Tujuan dari tahapan ini untuk mendapatkan informasi lebih lenjut tentang sistem yang berhasil di *exploit*. Dari hasil ini kita bisa mencari cara untuk mendapatkan akses ke internal organisasi,mengapus jejak,melakukan sniffing pada target tertentu, melihat data-data penting dan informasi-informasi penting lainnya dari sistem ke sistem yang telah kita *exploit*.

Dalam *Post Exploitation* ini kita bisa menilai, seberapa amankah sistem dari sebuah organisasi dimana penilain ini dihitung dari banyaknya infromasi yang didapatkan seperti data sensitive yang tersimpan pada sistem,mengidentifikasi konfigurasi yang salah, saluran komunikasi dan informasi lainnya. Namun pada tahapan ini, kita harus berhati-hati karena beberapa organisasi membatasi akses ke suatu informasi penting, sehingga selalu berdiskusi dengan klien akan memudahakan dalam proses dokumentasi.

g. Reporting

Reporting merupakan tahapan terakhir dalam PTES. Pada tahapan ini kita dituntu untuk membuat laporan untuk semua aktivitas yang kita lakukan, bagaimana kita melakukannya, dan yang terpenting bagaimana cara yang dilakukan oleh sebuah organisasi dalam mengamankan sistemnya selaman proses *penetration testing*.

2.2 Jenis Celah Keamanan Pada Website

Celah keamanan merupakan suatu kesalahan dalam pembuatan kode program dimana kesalahan ini dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab untuk melakukan tindakan illegal yang dapat merugikan pemilik program tersebut. Kerentanan bisa jadi bermacam-macam. Dalam website Exploit-DB, terdapat ratusan *exploit* yang diposting secara publik dengan tujuan agar sistem yang memilki kerentanan bisa segera diperbaiki (*patch*). Dari berbagai macam exploit yang tersebar di Internet terdapat beberapa kerentanan atau vulnerability yang sering terjadi khususnya pada website seperti SQL Injection, XSS, Default Password, Full Path Disclosure dan lain-lain.

2.2.1 Data Sensitif Pada File Robots

Robots Exlusion Procotol (ERP) atau biasa dikenal dengan robots.txt merupakan sebuah file teks yang digunakan pada website untuk menginstruksikan kepada search engine mengenai hal yang dapat di crawling dan dilakukan index di halaman search engine. Sebagai contoh google yang memiliki bot untuk menelusuri berbagai macam website, hasil penelusuran kemudian disimpan pada database google sehingga terkadang beberapa direktori yang harusnya bersifat rahasia dapat diakses dari informasi penelusuran search engine.

Melihat permasalah tersebut digunakan ERP sebagai solusi agar file dan dikektori sensitif tidak dapat ditelusuri oleh search engine (SANS Institute, 2011). Berikut adalah contoh penggunaan ERP untuk memberikan instruksi kepada semua search engine agar tidak melakukan penelusuran pada semua konten:

```
User-agent: *
Disallow: /
```

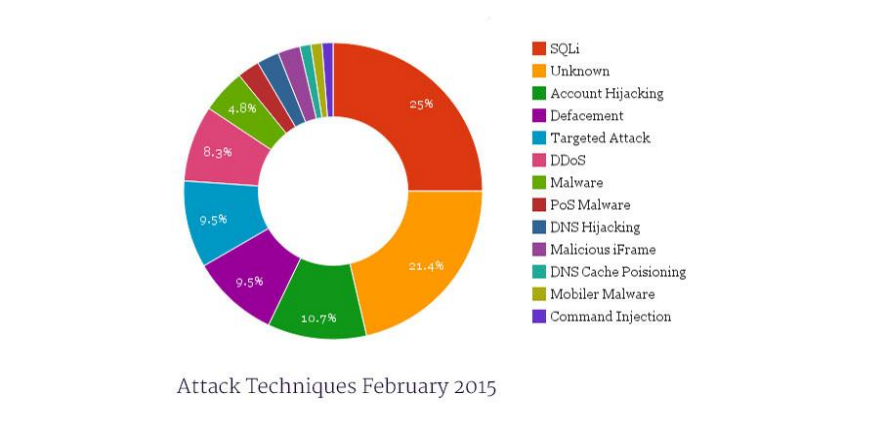
Contoh lainnya untuk memeberikan instruksi pada beberapa direktori:

```
User-agent: *
Disallow: /wp-content/uploads/
```

Dikarenakan file robots dapat diakses oleh siapapun, terdapat beberapa kasalahan yang dapat dilakukan oleh administrator seperti menginputkan folder rahasia yang dapat menjadi tambahan informasi bagi *hacker* untuk menjalankan aksi *hacking.*

2.2.2 SQL Injection

SQL Injection merupakan teknik dalam *hacking* dimana *attacker* dapat melakukan eksekusi perintah SQL pada website melalui input data dari *client* ke aplikasi web. Ketika seorang *attacker* berhasil melakukan teknik ini maka hal yang dapat dilakukan oleh attacker adalah membaca data penting pada *database* seperti *username* dan *password*, melakukan modifikasi *database*,mendownload semua data pada database dan hal lainnya yang berhubungan dengan manajemen database.



Gambar 2.1 Statistik Celah Keamanan

Sumber: [Optimizer WP, 2015]

SQL Injection merupakan teknik web hacking yang sangat populer,teknik ini pada tahun 2013 menjadi peringkat utama dalam statistik OWASP sebagai celah keamanan yang paling banyak ditemukan pada website yang berada di internet. Pada tahun 2015, celah ini masih menjadi yang tertinggi seperti pada **Gambar 2.1** yang di posting oleh Optimizer.

2.2.3 Cross Site Scripting

XSS merupakan salah satu jenis serangan injeksi code (code injection attack). XSS dilakukan oleh penyerang dengan cara memasukkan kode HTML atau client script code lainnya ke suatu situs. Serangan ini akan seolah-olah datang dari situs tersebut. Akibat serangan ini antara lain penyerang dapat mem-bypass keamanan di sisi klien, mendapatkan informasi sensitif, atau menyimpan aplikasi berbahaya (Lekies, Stock, & Johns, 2014). Terdapat dua jenis XSS, yaitu Stored XSS (Persistent) dan Reflected XSS (Non-Persistent)

- Stored XSS (Persistent)

Stored XSS merupakan serangan dimana script diinputkan secara permanen dan tersimpan di server target, seperti di dalam database website, buku tamu, komentar dan lain-lain. Stored XSS sangat berbahaya, karena seorang attacker bisa membuat halaman palsu dan dapat menipu user yang mengunjungi website yang sudah terinjeksi.

- Reflected XSS (Non-Persistent)

Refrected XSS merupakan serangan dimana *script* juga diinputkan namun tidak secara permanen atau tidak tersimpan di server target, *attacker* hanya melakukan injeksi pada URL dari sebuah website misalnya pada hasil pencarian atau inputan yang tidak memiliki validasi. Pada Reflected XSS, attacker biasanya melakukan teknik Social Engineering dengan menyebarkan URL ini melaui Social Media, blog, e-mail dan lain-lain. Dengan cara ini attacker dapat mencuri cookies user yang mengunjungi URL tersebut.

2.2.4 Full Path Disclosure

Full Path Disclosure merupakan celah keamanan dimana attacker dapat melihat path direktori dari sebuah website.Resiko yang ditimbulkan dari celah ini adalah ketika hacker berhasil menemukan celah misalnya *Local File Include* maka *attacker* dapat melakukan injeksi untuk mengakses file-file penting sesuai dengan informasi path yang diapatkan. Contoh Full Path Disclosure:

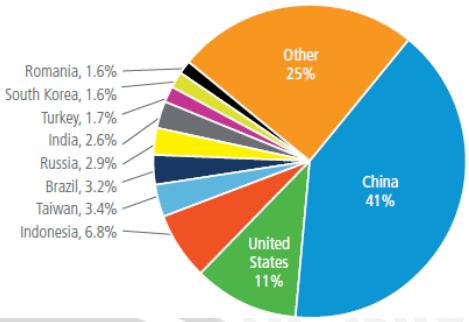
```
Warning: session_start() [function.session-start]: The
session id contains illegal characters,
valid characters are a-z, A-Z, 0-9 and '-,' in
/home/alice/public_html/includes/functions.php on line 2
```

Gambar 2.2 Contoh Full Path Disclosure
Sumber:[OWASP, 2012]

2.3 Statistik Serangan

Pada tahun 2014, Akamai membuat statistik laporan serangan internet dunia, dan yang mengejutkan Indonesia berada di urutan ke-3 setelah China dan Amerika dalam hal penyerangan di internet. Sebelumnya pada tahun 2013, Indonesia melakukan penyerangan sebanyak 5.7% dan pada tahun 2014 naik menjadi 6.8%.Hal ini tentunya harus menjadi perhatian bagi Pemerintah di Indonesia terkait ancaman dalam negeri.

	Country/Region	Q1 '14 Traffic %	Q4 '13 %
1	China	41%	43%
2	United States	11%	19%
3	Indonesia	6.8%	5.7%
4	Taiwan	3.4%	3.4%
5	Brazil	3.2%	1.1%
6	Russia	2.9%	1.5%
7	India	2.6%	0.7%
8	Turkey	1.7%	0.4%
9	South Korea	1.6%	0.6%
10	Romania	1.6%	0.9%
–	Other	25%	12%

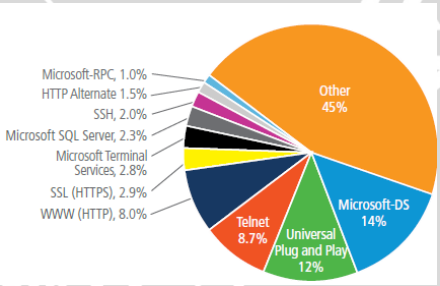


Gambar 2.3 Statistik Serangan Dunia Tahun 2014

Sumber: [Akamai, 2014]

Selain informasi dari berbagai macam Negara, terdapat juga statistik yang dipublikasikan oleh Akamai, yaitu tentang port pada server yang sering diserang dan yang menjadi fokus pada topik skripsi ini adalah port 80 dimana port ini digunakan oleh Apache Web Server untuk menampilkan halaman website. Port ini masuk dalam 10 port yang paling sering diserang.

Port	Port Use	Q1 '14 Traffic %	Q4 '13 %
445	Microsoft-DS	14%	30%
5000	Universal Plug & Play	12%	<0.1%
23	Telnet	8.7%	3.0%
80	WWW (HTTP)	8.0%	14%
443	SSL (HTTPS)	2.9%	8.2%
3389	Microsoft Terminal Services	2.8%	4.9%
1433	Microsoft SQL Server	2.3%	4.9%
22	SSH	2.0%	3.6%
8080	HTTP Alternate	1.5%	2.7%
135	Microsoft-RPC	1.0%	2.0%
Various	Other	45%	–



Gambar 2.4 Port Paling Sering Diserang

Sumber: [Akamai, 2014]

2.4 Pengelolaan Website Di Universitas Brawijaya

Universitas Brawijaya memiliki domain ub.ac.id. Domain ini dapat juga digunakan sebagai subdomain bagi lembaga/organisasi yang ingin membuat hosting.Layanan hosting ini disediakan oleh PPTI secara gratis.Adapun Lembaga/Organisasi yang dapat membuat hosting adalah sebagai berikut :

- Fakultas
- Jurusan
- Lembaga Universitas/Fakultas
- Lembaga Kemahasiswaan
- Himpunan Mahasiswa
- Unit Aktivitas Mahasiswa

Dalam pembuatan hosting, lembaga/organisasi harus mengajukannnya melalui PPTI dengan membawa persyaratan-persyaratan yang tertera di website hosting.ub.ac.id. Pihak PPTI hanya meyediakan layanan hosting dan untuk administrasi/pengelolaan hosting akan diserahkan kepada lembaga/organisasi. Kesimpulannya adalah bahwa setiap website dikelola oleh admin yang berbeda-

beda dan jika terdapat sebuah celah keamanan, maka yang bertanggung jawab adalah pengelola website karena PPTI hanya menyediakan layanan hosting tanpa adanya sistem pendeteksi celah keamanan pada setiap website yang menggunakan domain ub.ac.id.

2.5 WPSCAN

WPScan merupakan tool hacking yang khusus digunakan untuk melakukan scanning pada website yang menggunakan WordPress. WPScan bekerja dengan menggunakan metode *black-box scanning,* artinya proses *scanning* dilakukan secara langsung pada target tanpa menggunakan pihak ketiga seperti *search engine* dan *whois*.

Dalam proses *scanning*, WPScan menggunakan database yang bersumber dari website *WPScan Vulnerability Database* (http://wpvulndb.com). Website tersebut berisi celah keamanan yang terbagi tiga kategori yaitu:

1. Celah keamanan pada versi WordPress
2. Celah keamanan pada *plugin* WordPress
3. Celah keamanan pada *template* WordPress

Adapun fitur lainnya yang terdapat pada WPScan adalah melakukan deteksi celah keamanan seperti *Full Path Disclosure*, *username default*, dan *directory listing.* Semua data celah keamanan yang digunakan oleh WPScan selalu diperbaharui ketika terdapat *update* pada website *WPScan Vulnerability Database.*

2.6 Uniscan

Uniscan merupakan tool yang digunakan untuk mendeteksi celah keamanan pada website.Uniscan sudah tersedia pada Kali Linux versi terbaru dan dapat langsung digunakan. Uniscan dapat dijalankan via terminal maupun via *interface GUI* Adapun fitur yang terdapat pada Uniscan adalah sebagai berikut:

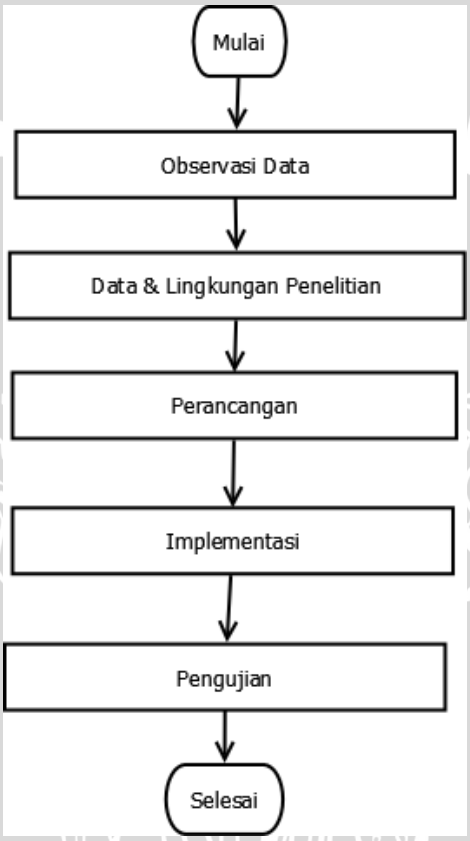
1. Deteksi Email
2. URL Crawling, yaitu mengumpulkan semua URL yang akan dijadikan sebagai target *scanning*
3. Deteksi *SQL Injection*
4. Deteksi *Cross Site Scripting*
5. Deteksi celah *Local File Include*
6. Deteksi celah *Remote Command Execution*

BAB 3 METODOLOGI

Metodologi penelitian berisi tentang metode yang digunakan untuk melakukan penelitian atau dapat juga diartikan tindakan yang digunakan untuk teknik pemilihan rancangan di dalam penelitian ini.

3.1 Tahapan Penelitian

Tahapan yang dilakukan dalam penelitian ini dapat dilihat pada **Gambar 3.1** sebagai berikut:



Gambar 3.1 Alur Penelitian

Berdasarkan bagan pada **Gambar 3.1**, tahapan penelitian dapat dijelaskan sebagai berikut:

1. Melakukan observasi untuk mendapatkan data-data dalam menunjang penelitian
2. Melakukan pemilihan data dan lingkungan penelitian
3. Melakukan perancangan sistem sesuai data dan lingkungan penelitian
4. Melakukan implementasi sistem berdasarkan data pada perancangan.
5. Melakukan pengujian dan analisis pada sistem yang telah dikembangkan.

3.2 Observasi Data

Observasi bertujuan untuk mengumpulkan data yang digunakan dalam menunjang penelitian.Data ini menentukan bagaimana sistem akan dibangun beserta teknik apa yang akan digunakan dalam mendeteksi celah keamanan. Adapun data yang dikumpulkan dalam observasi ini adalah sebagai berikut:

1. Daftar website-website di Universitas Brawijaya dengan *domain* utama ub.ac.id dan *domain* pada masing-masing unit yang menggunakan *subdomain* .ub.ac.id.
2. Jenis website yang digunakan apakah menggunakan Content Management System (CMS) atau menggunakan PHP manual/framework. Tujuan pendeteksian ini adalah untuk melihat jenis website apakah yang paling banyak digunakan di Universitas Brawijaya sehingga dapat penulis dapat membuat sistem yang lebih spesifik dalam mendeteksi celah keamanan pada jenis website tertentu.
3. Informasi celah keamanan sesuai dengan jenis website yang ditemukan, misalnya jika website terdeteksi menggunakan WordPress, maka akan dikumpulkan informasi celah keamanan yang berhubungan dengan WordPress dan begitu seterusnya tergantung pada jenis website yang ditemukan.

Ketiga data diatas dikumpulkan secara berurut mulai dari pengumpulan data website hingga informasi celah keamanan pada jenis website tertentu.Informasi celah keamanan pada jenis website tidak bisa dilakukan bila jenis website yang digunakan belum diketahui dan untuk mengetahui jenis website tidak bisa dilakukan jika data-data pada semua domain belum dikumpulkan.

Dalam penelitian ini, diperlukan teknik yang digunakan untuk mengumpulkan semua data-data yang dibutuhkan. Adapun teknik pengumpulan data yang akan digunakan dalam observasi ini adalah sebagai berikut:

1. Data berupa daftar nama-nama domain di Universitas Brawijaya akan dikumpulkan dengan memanfaatkan *search engine* seperti Google, Yahoo, Bing dan Netcraft. Pengumpulan data dapat dilkakukan secara otomatis dengan menggunakan *tools* recon-ng. Pada recon-ng terdapat fitur yang secara umum bekerja untuk mencari informasi nama domain beserta subdomain berdasarkan dari *search engine* yang telah disebutkan sebelumnya, kemudian hasil pengumpulan data akan disimpan dalam bentuk laporan dengan tipe file html.
2. Setelah mendapatkan daftar nama-nama domain, selanjutnya masing-masing domain diteliti untuk menentukan jenis website yang digunakan. Dalam penelitian diberikan batasan penelitian dimana pendeteksian akan dilakukan pada dua jenis website yaitu CMS dan PHP manual/framework. CMS merupakan website dengan banyak jenis seperti WordPress, Joomla, Drupal dan lain-lain, untuk dapat mendeteksi jenis CMS yang digunakan, penulis menggunakan Guess CMS Detector (http://guess.scritch.org). Untuk PHP manual / framework tidak memiliki kriteria khusus dikarenakan dalam pengembangannya yang dilakukan dari awal, tidak seperti CMS yang sudah

tersedia dan dapat langsung digunakan sehingga dalam peneltian ini jika sebuah webite tidak terdeteksi menggunakan CMS maka akan dianggap menggunakan PHP manual / *framework*.

3. Informasi celah keamanan berdasarkan jenis website dilakukan bilamana semua jenis-jenis website sudah ditentukan. Pencarian informasi celah keamanan akan menjadi acuan dalam membangun sistem mengenai teknik apa yang digunakan pada sistem untuk mendeteksi celah keamanan pada website. Informasi celah keamanan akan difokuskan pada jenis website CMS dan PHP manual / framework.

3.3 Data & Lingkungan Penelitian

Website yang berada di Universitas Brawijaya memiliki banyak domain yang disediakan untuk masing-masing.Dalam penelitian ini dilakukan simulasi dimana sistem yang dibangun kemudian diuji pada website-website di Universitas Brawijaya, karena pengujian yang bersifat simulasi maka dibutuhkan data-data yang valid mengenai website yang akan dijadikan sebagai data pengujian.

Data & lingkungan penelitian mencakup data hasil observasi mulai dari data domain, jenis website hingga informasi celah keamanan, data ini kemudian diolah kembali dengan tujuan untuk membuat replikasi website sehingga data tersebut dapat dijadikan sebagai data pengujian yang digunakan untuk simulasi penyerangan pada website. Semua data pengujian dibuat secara *offline* pada komputer lokal peneliti sehingga dalam pengujiannya juga akan dilakukan secara *offline*.

Data yang telah diolah selanjutnya dapat digunakan untuk membuat lingkungan penelitian guna mendukung proses simulasi. Lingkungan penelitian yang dibahas dalam penelitian ini adalah pembuatan dan pemetaan arsitektur jaringan di Universitas Brawijaya khususnya dalam manajemen website.

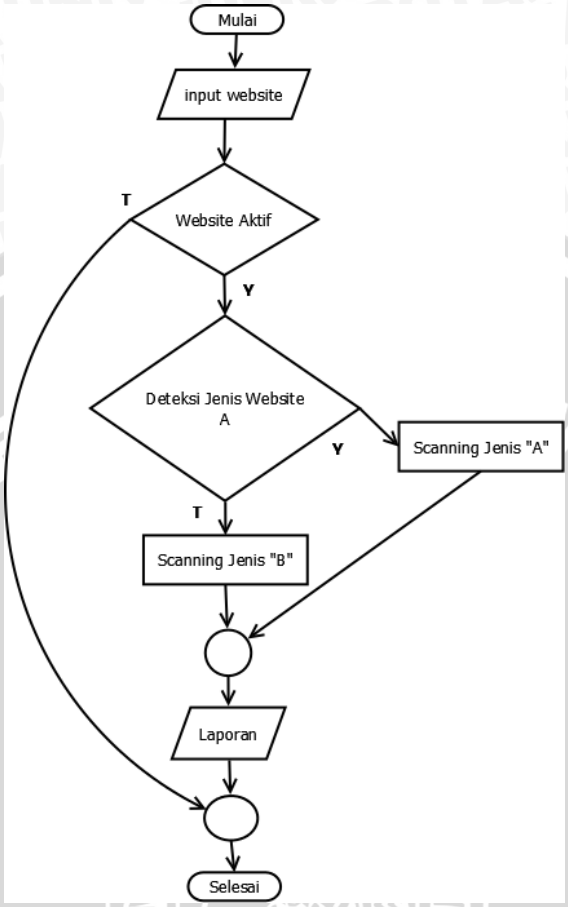
3.4 Perancangan

Pada tahap perancangan, penelitii melakukan beberapa perancangan meliputi gambaran umum sistem yang menjelaskan cara kerja sistem secara umum, alur kerja sistem dimana menjelaskan cara kerja sistem dan selanjutnya akan dibahas mengenai pengembangan sistem sesuai dengan teknik yang telah ditentukan dalam mencari celah keamanan.

Perancangan sistem sangat bergantung dari hasil data & lingkungan penelitian. Sistem deteksi celah keamanan yang dikembangkan melakukan deteksi celah keamanan berdasarkan jenis website, contohnya jenis CMS dan PHP manual / framework. Untuk website CMS membutuhkan penelitian lebih lanjut mengenai jenis CMS yang digunakan apakah menggunakan WordPress, Joomla, Drupal atau jenis lainnya.

Pemilihan website CMS ditentukan dari seberapa banyaknya data yang ditemukan atau data website CMS yang dominan akan menjadi acuan dalam membuat sistem,misalnya jika website CMS WordPress lebih banyak ditemukan

daripada website CMS Joomla maka akan dipilih WordPress sebagai acuan dalam membuat sistem dan begitu sebaliknya. Adapun gambaran umum sistem yang akan dibangun dapat dilihat pada **Gambar 3.2**.



Gambar 3.2 Flowchart Gambaran Umum Sistem

Berdasarkan **Gambar 3.2**, user memasukkan URL berupa domain yang akan dijadikan sebagai website target *scanning*. URL yang telah dimasukkan kemudian dilakukan pengecekan apakah website aktif atau tidak, jika tidak maka proses *scanning* selesai namun jika aktif maka sistem melanjutkan pada tahap pendeteksian jenis website, jika website terdeteksi menggunakan jenis website "A", maka lakukan deteksi celah keamanan sesuai dengan teknik *scanning* pada website jenis "A". Jika website "A" tidak terdeteksi maka website akan melakukan scanning pada website jenis "B" hingga akhirnya semua hasil *scanning* akan ditampilkan dalam bentuk laporan berupa celah kemanan yang ditemukan.

Sistem deteksi celah keamanan yang dirancang dalam penelitian ini dikembangkan dengan bahasa pemrograman Python yang kemudian file python ini dijalankan via website dengan tujuan agar dapat diakses oleh semua pengguna tidak terbatas pada sistem operasi yang digunakan. Website dalam hal ini hanya digunakan sebagai perantara dalam eksekusi file saja, namun inti dari sistem ini sendiri adalah file python yang berfungsi melakukan scanning secara keseluruhan.

3.5 Implementasi

Implementasi merupakan tindakan lanjutan atau penerapan dari perancangan yang sudah disusun sebelumnya. Dalam penelitian ini yang diimplementasikan adalah teknik scanning yang digunakan dalam mendeteksi celah keamanan pada jenis website tertentu yang dibuat dengan bahasa pemrograman python.Selanjutnya dilakukan implementasi pada website yang digunakan sebagai perantara untuk melakukan eksekusi *file* python tersebut.

3.6 Pengujian

Dalam tahap pengujian, dilakukan simulasi untuk mengukur performa dari sistem yang telah dikembangkan.Adapun yang menjadi alat ukur dalam proses uji coba ini meliputi pendeteksian celah keamanan dan akurasi sistem.Untuk prosedur pengujian yang dilakukan adalah sebagai berikut.

3.6.1 Pengujian Deteksi Celah Keamanan

Pengujian deteksi celah keamanan bertujuan untuk menguji sistem dalam mendeteksi celah keamanan pada data uji yang disediakan. Dengan dilakukannya pengujian ini hasil yang diharapkan adalah data berupa laporan statistik celah keamanan yang ditemukan pada semua website di Universitas Brawijaya, laporan ini dapat digunakan untuk memberikan informasi kepada pihak Universitas Brawijaya mengenai celah keamanan apa saja yang ditemukan serta solusi untuk mengatasi celah keamanan tersebut.

Prosedur pengujian deteksi celah keamanan dilakukan dengan menjalankan sistem pada semua data uji. Hasil *scanning* kemudian dilakukan analisis dari semua data uji ada berapa yang terdeteksi memiliki celah keamanan dan untuk mengetahui celah keamanan yang dominan.

Dalam pengujian ini diperlukan tahapan pengujian, agar proses dalam pengujian sistem dapat terlaksana dengan baik. Adapun tahapan pengujian yang dilakukan adalah sebagai berikut:

1. Mempersiapkan semua data pengujian.
2. Melakukan deteksi celah keamanan pada semua data uji.
3. Melakukan analisis pada hasil yang didapatkan, analisis yang dilakukan bertujuan untuk mendapatkan celah yang paling banyak ditemukan pada website-website di Universitas Brawijaya.
4. Dari hasil analaisis, selanjutnya membuat grafik celah keamanan yang dominan, tujuan dari pembuatan grafik adalah untuk mempresentasikan celah keamanan yang dominan khususnya pada website-website di Universitas Brawijaya.

3.6.2 Pengujian Akurasi Sistem

Pada pengujian ini dilakukan percobaan untuk menguji akurasi sistem, adapun tujuan pengujian akurasi sistem ini adalah untuk mengetahui seberapa efektif sistem yang telah dikembangkan jika dibandingkan dengan *tools* lainnya.

Prosedur pengujian akurasi sistem dilakukan dengan melakukan *scanning* pada website-website yang berada di Universitas Brawijaya. Masing-masing website di uji dengan menggunakan sistem deteksi celah keamanan beserta *tools vulnerability scanner* lainnya, terdapat tiga *tools* yang digunakan untuk melakukan pengujian dimana hasil pengujian dari ketiga *tools* ini dibandingkan dengan sistem yang telah dikembangkan oleh penulis. Berikut adalah *tools* yang digunakan:

- Nessus
- Acunetix Web Vulnerability Scanner
- Nikto

Pemilihan ketiga tools diatas berdasarkan data dari website SecTools dimana artikel pada web tersebut membahas *tools vulnerability scanner* yang paling sering digunakan.

Setelah memilih *tool* yang digunakan untuk pengujian, selanjutnya dilakukan pemilihan data uji. Website yang diuji dipilih secara acak dari data replikasi website yang telah didapatkan. Adapun tahapan pengujian yang dilakukan pada pengujian ini adalah sebagai berikut.

1. Mempersiapkan data uji.
2. Melakukan deteksi celah keamanan pada semua data uji mulai dari mengggunakan tool Nessus, Nikto, Acunetix dan sistem deteksi celah keamanan yang dikembangkan oleh peneliti.
3. Melakukan analisis terkait celah keamanan yang ditemukan oleh masing-masing *tools* dan menampilkan hasil analisis berupa tabel celah keamanan yang ditemukan oleh masing-masing *tools*.

BAB 4 DATA DAN LINGKUNGAN PENELITIAN

Pada Bab 4 ini membahas hasil observasi data yang telah didapatkan kemudian diolah kembali untuk dijadikan data pengujian. Selanjutnya dibahas mengenai lingkungan penelitian dan lingkungan pengujian.

4.1 Hasil Observasi Data

4.1.1 Data Domain Website

Data domain website dikumpulkan dengan melakukan pencarian data melalui *search engine* yang dalam hal ini memanfaatkan fitur pada tool recon-ng yang mencari domain berdasarkan Google, Yahoo, dan Bing. Berikut adalah daftar hasil domain yang ditemukan:

Tabel 4.1 Daftar Domain Website

No	URL
1	http://feb.ub.ac.id
2	http://apcaf.ub.ac.id
3	http://bet.ub.ac.id
4	http://fp.ub.ac.id
5	http://accounting.feb.ub.ac.id
6	http://apis.ub.ac.id
7	http://bak.ub.ac.id
8	http://arsitektur.ub.ac.id
9	http://basicsc.ub.ac.id/
10	http://bakp.ub.ac.id
11	http://bicab.ub.ac.id
12	http://biologi.ub.ac.id
13	http://biosains.ub.ac.id/
14	http://blog.ub.ac.id
15	http://bits.ub.ac.id
16	http://bss.ub.ac.id
17	http://sd.bss.ub.ac.id
18	http://smp.ub.ac.id
19	http://ppsub.ub.ac.id/
20	http://ppti.ub.ac.id
21	http://eeccis.ub.ac.id/2016/
22	http://em.ub.ac.id
23	http://lib.ub.ac.id
24	http://fapet.ub.ac.id
25	http://teknikkimia.ub.ac.id
26	http://fia.ub.ac.id
27	http://fisip.ub.ac.id

28	http://griya.ub.ac.id
29	http://fpik.ub.ac.id
30	http://hme.ub.ac.id
31	http://hukum.ub.ac.id
32	http://icambbe.ub.ac.id/
33	http://icoee2015.ub.ac.id/
34	http://icon.ub.ac.id
35	http://ie.feb.ub.ac.id
36	http://isyg2015.ub.ac.id/
37	http://industri.ub.ac.id
38	http://io.ub.ac.id
39	http://upkk.ub.ac.id
40	http://kepegawaian.ub.ac.id/
41	http://kerjasama.ub.ac.id/
42	http://kimia.ub.ac.id
43	http://lab-srk.ub.ac.id/
44	http://lpke.ub.ac.id/
45	http://lp3.ub.ac.id
46	http://mansyla.ub.ac.id/
47	http://mipa.ub.ac.id
48	http://mku.ub.ac.id
49	http://pdg.fk.ub.ac.id/
50	http://pemerintahan.ub.ac.id/
51	http://pemira.ub.ac.id
52	http://pengairan.ub.ac.id/
53	http://pilrek2014.ub.ac.id/
54	http://pjm.ub.ac.id
55	http://pkkss.ub.ac.id
56	http://ppikid.ub.ac.id
57	http://ppak.feb.ub.ac.id
58	http://pwk.ub.ac.id
59	http://selma.ub.ac.id
60	http://singo.ub.ac.id/
61	http://sipil.ub.ac.id/
62	http://soi2015.ub.ac.id/
63	http://sosek.ub.ac.id/
64	http://staff.ub.ac.id
65	http://spi.ub.ac.id
66	http://tanah.ub.ac.id
67	http://teknik.ub.ac.id
68	http://tip.ub.ac.id
69	http://tp.ub.ac.id
70	http://uaki.ub.ac.id

71	http://ulp.ub.ac.id
72	http://vokasi.ub.ac.id
73	http://poros.ub.ac.id
74	http://display.ub.ac.id
75	http://krismatiik.ub.ac.id/
76	http://thp.ub.ac.id
77	http://optiik.ub.ac.id
78	http://mbesb.ub.ac.id
79	http://rajabrawijaya.ub.ac.id
80	http://mesin.ub.ac.id
81	http://fp.ub.ac.id/
82	http://elektro.ub.ac.id/lab-dpk
83	http://teknik.ub.ac.id/dosen/
84	http://filkom.ub.ac.id
85	http://sikma.ub.ac.id
86	http://amd.ub.ac.id
87	http://bemtiik.ub.ac.id
88	http://hmif.ub.ac.id
89	http://guesthouse.ub.ac.id
90	http://basic-registration.ub.ac.id
91	http://karyailmiah.fp.ub.ac.id
92	http://aec.fia.ub.ac.id
93	http://apatte62.ub.ac.id
94	http://iorc.ub.ac.id
95	http://fib.ub.ac.id

Berdasarkan tabel 4.1, ditemukan 95 website dengan subdomain ub.ac.id. Pada hasil observasi ini, penulis masih memliliki keterbatasan penggalian informasi dimana hanya website yang berada pada *search engine* saja yang didapatkan.

Dari 95 data, dilakukan penelurusan lebih lanjut mengenai apakah website dapat diakses (aktif) atau tidak (tidak aktif). Tabel 4.2 memperlihat daftar website yang tidak dapat diakses atau statusnya saat ini down. Website-website ini diakses pada 25 juni 2016.

Tabel 4.2 Website tidak aktif

1	http://iorc.ub.ac.id
2	http://ppak.feb.ub.ac.id
3	http://pemira.ub.ac.id
4	http://isyg2015.ub.ac.id

Terdapat 3 website yang tidak bisa diakses sehingga total data penelitian yang akan digunakan adalah berjumlah 91 domain.

4.1.2 Jenis Website

Data domain website yang telah dikumpulkan selanjutnya dilakukan pendeteksian jenis website dengan tujuan agar pembuatan sistem dalam penelitian ini berfokus pada jenis website yang ditemukan.

Penentuan jenis website dilakukan dengan memfokuskan pendeteksian pada website jenis CMS dimana pendeteksian dilakukan menggunakan Guess CMS Detector. Jika CMS tidak terdeteksi, maka website dianggap menggunakan PHP Manual / framework.

Berdasarkan hasil pendeteksian, dari 91 data domain ditemukan 73 website menggunakan website jenis WordPress dan 18 website menggunakan PHP manual / framework, berikut adalah hasil pendeteksian yang telah didapatkan :

Tabel 4.3 Hasil Pendeteksian Jenis Website

No	URL	Jenis Website
1	http://feb.ub.ac.id	WordPress
2	http://apcaf.ub.ac.id	WordPress
3	http://bet.ub.ac.id	WordPress
4	http://fp.ub.ac.id	WordPress
5	http://apis.ub.ac.id	WordPress
6	http://bak.ub.ac.id	WordPress
7	http://arsitektur.ub.ac.id	WordPress
8	http://basicsc.ub.ac.id/	WordPress
9	http://bakp.ub.ac.id	WordPress
10	http://bicab.ub.ac.id	WordPress
11	http://biologi.ub.ac.id	WordPress
12	http://biosains.ub.ac.id/	WordPress
13	http://blog.ub.ac.id	WordPress
14	http://bits.ub.ac.id	WordPress
15	http://bss.ub.ac.id	WordPress
16	http://ppsub.ub.ac.id/	WordPress
17	http://ppti.ub.ac.id	WordPress
18	http://eeccis.ub.ac.id/2016/	WordPress
19	http://em.ub.ac.id	WordPress
20	http://lib.ub.ac.id	WordPress
21	http://fapet.ub.ac.id	WordPress
22	http://teknikkimia.ub.ac.id	WordPress
23	http://fia.ub.ac.id	WordPress
24	http://fisip.ub.ac.id	WordPress
25	http://fpik.ub.ac.id	WordPress
26	http://hme.ub.ac.id	WordPress
27	http://hukum.ub.ac.id	WordPress
28	http://icambbe.ub.ac.id/	WordPress

29	http://icoee2015.ub.ac.id/	WordPress
30	http://icon.ub.ac.id	WordPress
31	http://ie.feb.ub.ac.id	WordPress
32	http://industri.ub.ac.id	WordPress
33	http://io.ub.ac.id	WordPress
34	http://upkk.ub.ac.id	WordPress
35	http://kepegawaian.ub.ac.id/	WordPress
36	http://kerjasama.ub.ac.id/	WordPress
37	http://kimia.ub.ac.id	WordPress
38	http://lab-srk.ub.ac.id/	WordPress
39	http://lpke.ub.ac.id/	WordPress
40	http://lp3.ub.ac.id	WordPress
41	http://mansyla.ub.ac.id/	WordPress
42	http://mipa.ub.ac.id	WordPress
43	http://mku.ub.ac.id	WordPress
44	http://pdg.fk.ub.ac.id/	WordPress
45	http://pemerintahan.ub.ac.id/	WordPress
46	http://pengairan.ub.ac.id/	WordPress
47	http://pilrek2014.ub.ac.id/	WordPress
48	http://pjm.ub.ac.id	WordPress
49	http://pkkss.ub.ac.id	WordPress
50	http://ppikid.ub.ac.id	WordPress
51	http://pwk.ub.ac.id	WordPress
52	http://selma.ub.ac.id	WordPress
53	http://singo.ub.ac.id/	WordPress
54	http://sipil.ub.ac.id/	WordPress
55	http://soi2015.ub.ac.id/	WordPress
56	http://sosek.ub.ac.id/	WordPress
57	http://staff.ub.ac.id	WordPress
58	http://spi.ub.ac.id	WordPress
59	http://tanah.ub.ac.id	WordPress
60	http://teknik.ub.ac.id	WordPress
61	http://tip.ub.ac.id	WordPress
62	http://tp.ub.ac.id	WordPress
63	http://uaki.ub.ac.id	WordPress
64	http://ulp.ub.ac.id	WordPress
65	http://vokasi.ub.ac.id	WordPress
66	http://poros.ub.ac.id	WordPress
67	http://display.ub.ac.id	WordPress
68	http://krismatiik.ub.ac.id/	WordPress
69	http://thp.ub.ac.id	WordPress
70	http://optiik.ub.ac.id	WordPress
71	http://aec.fia.ub.ac.id	WordPress

72	http://apatte62.ub.ac.id	WordPress
73	http://fib.ub.ac.id	WordPress
74	http://mbesb.ub.ac.id	PHP manual/framework
75	http://rajabrawijaya.ub.ac.id	PHP manual/framework
76	http://mesin.ub.ac.id	PHP manual/framework
77	http://fp.ub.ac.id/	PHP manual/framework
78	http://elektro.ub.ac.id/lab-dpk	PHP manual/framework
79	http://teknik.ub.ac.id/dosen/	PHP manual/framework
80	http://filkom.ub.ac.id	PHP manual/framework
81	http://sikma.ub.ac.id	PHP manual/framework
82	http://amd.ub.ac.id	PHP manual/framework
83	http://bemtiik.ub.ac.id	PHP manual/framework
84	http://hmif.ub.ac.id	PHP manual/framework
85	http://guesthouse.ub.ac.id	PHP manual/framework
86	http://basic-registration.ub.ac.id	PHP manual/framework
87	http://karyailmiah.fp.ub.ac.id	PHP manual/framework
88	http://smp.ub.ac.id	PHP manual/framework
89	http://sd.bss.ub.ac.id	PHP manual/framework
90	http://griya.ub.ac.id	PHP manual/framework
91	http://accounting.feb.ub.ac.id	PHP manual/framework

Dari hasil pendeteksian ini, penulis memiliki hipotesis bahwa website yang berada di Universitas Brawijaya lebih dominan menggunakan CMS Wordpress. Karena jenis website yang dominan ini, penulis kemudian melakukan penelitian lebih lanjut mengenai versi WordPress yang digunakan dimana pendeteksian versi ini kemudian dapat menjadi acuan peneliti dalam membuat replikasi website.

Dalam melakukan pendeteksian versi WordPress, peneliti menggunakan tool otomatisi yang dikembangkan menggunakan Python. Berikut adalah source code dari tool tersebut:

```
#!/usr/bin/python
import re,urllib2,requests
from BeautifulSoup import BeautifulSoup
with open('url.txt','r') as myfile:
    data=myfile.read().split()
    for lines in range(len(data)):
        url=data[lines]
        folderCheck=requests.get("%s/wp-content" % (url))
        folderCheck2=requests.get("%s/wp-login.php" % (url))
        if folderCheck or folderCheck2:
            html_page=urllib2.urlopen(url)
            soup=BeautifulSoup(html_page)
            try:
                #cek meta content apakah berisi string WordPress
```

```
hasil=str(soup.first('meta',attrs={'content':re.compile("WordPress")})['content'])
            except:
                print url
                print "Tidak dapat mendeteksi versi WordPress\n"
            else:
                print url
                print "Tipe Website :",hasil,"\n"
        else:
            print url
            print "Tipe Website : PHP/Framework\n"
```

Gambar 4.1 Source Code Deteksi Versi WordPress

Cara kerja tool ini adalah mendeteksi apakah terdapat file wp-login.php atau direktori wp-content, jika salah satunya ditemukan selanjutkan akan dilakukan pengecekan apakah terdapat kata "WordPress", jika ditemukan maka *tool* akan menyimpulkan bahwa website tersebut menggunakan WordPress dan akan mengambil informasi versi WordPress yang digunakan pada *tag meta*. Jika pada *tag meta* tidak ditemukan, maka versi WordPress tidak dapat ditemukan (undetected).

Dari hasil pendeteksian pada 73 website yang menggunakan WordPress ditemukan 56 versi yang terdeteksi sedangkan 17 website lainnya tidak terdeteksi versi berapa yang digunakan. Hasil pendeteksian versi WordPress dapat dilihat pada **Tabel 4.4**

Tabel 4.4 Hasil Deteksi Versi WordPress

No	URL	Jenis Website	Versi
1	http://feb.ub.ac.id	WordPress	Undetected
2	http://apcaf.ub.ac.id	WordPress	3.8.2
3	http://bet.ub.ac.id	WordPress	4.1.8
4	http://fp.ub.ac.id	WordPress	Undetected
5	http://apis.ub.ac.id	WordPress	3.5.1
6	http://bak.ub.ac.id	WordPress	3.8.1
7	http://arsitektur.ub.ac.id	WordPress	3.5.1
8	http://basicsc.ub.ac.id/	WordPress	Undetected
9	http://bakp.ub.ac.id	WordPress	3.8
10	http://bicab.ub.ac.id	WordPress	3.4.2
11	http://biologi.ub.ac.id	WordPress	3.7.1
12	http://biosains.ub.ac.id/	WordPress	Undetected
13	http://blog.ub.ac.id	WordPress	Undetected
14	http://bits.ub.ac.id	WordPress	3.1.2
15	http://bss.ub.ac.id	WordPress	Undetected
16	http://ppsub.ub.ac.id/	WordPress	3.4.2
17	http://ppti.ub.ac.id	WordPress	3.8.1
18	http://eeccis.ub.ac.id/2016/	WordPress	3.8

19	http://em.ub.ac.id	WordPress	4.1.8
20	http://lib.ub.ac.id	WordPress	Undetected
21	http://fapet.ub.ac.id	WordPress	3.5.1
22	http://fib.ub.ac.id	WordPress	Undetected
23	http://fia.ub.ac.id	WordPress	4.4
24	http://fisip.ub.ac.id	WordPress	4.3.1
25	http://fpik.ub.ac.id	WordPress	3.8.1
26	http://hme.ub.ac.id	WordPress	Undetected
27	http://hukum.ub.ac.id	WordPress	3.8.1
28	http://icambbe.ub.ac.id/	WordPress	3.8.1
29	http://icoee2015.ub.ac.id/	WordPress	4.2.5
30	http://icon.ub.ac.id	WordPress	3.1
31	http://ie.feb.ub.ac.id	WordPress	Undetected
32	http://industri.ub.ac.id	WordPress	3.6
33	http://io.ub.ac.id	WordPress	4.2.5
34	http://upkk.ub.ac.id	WordPress	3.5.1
35	http://kepegawaian.ub.ac.id/	WordPress	4.2.2
36	http://kerjasama.ub.ac.id/	WordPress	3.8.1
37	http://kimia.ub.ac.id	WordPress	3.9-beta1
38	http://lab-srk.ub.ac.id/	WordPress	4.0
39	http://lpke.ub.ac.id/	WordPress	3.5.1
40	http://lp3.ub.ac.id	WordPress	Undetected
41	http://mansyla.ub.ac.id/	WordPress	Undetected
42	http://mipa.ub.ac.id	WordPress	3.7.1
43	http://mku.ub.ac.id	WordPress	Undetected
44	http://pdg.fk.ub.ac.id/	WordPress	3.5.2
45	http://pemerintahan.ub.ac.id/	WordPress	Undetected
46	http://pengairan.ub.ac.id/	WordPress	3.7.1
47	http://pilrek2014.ub.ac.id/	WordPress	3.8
48	http://pjm.ub.ac.id	WordPress	3.5.1
49	http://pkkss.ub.ac.id	WordPress	Undetected
50	http://ppikid.ub.ac.id	WordPress	4.0
51	http://pwk.ub.ac.id	WordPress	3.8.1
52	http://selma.ub.ac.id	WordPress	3.1.2
53	http://singo.ub.ac.id/	WordPress	3.8.1
54	http://sipil.ub.ac.id/	WordPress	3.6.1
55	http://soi2015.ub.ac.id/	WordPress	Undetected
56	http://sosek.ub.ac.id/	WordPress	Undetected
57	http://staff.ub.ac.id	WordPress	3.9.2
58	http://spi.ub.ac.id	WordPress	3.8.1
59	http://tanah.ub.ac.id	WordPress	3.5.2
60	http://teknik.ub.ac.id	WordPress	4.3.1
61	http://tip.ub.ac.id	WordPress	3.4.2

62	http://tp.ub.ac.id	WordPress	3.7.1
63	http://uaki.ub.ac.id	WordPress	4.3.1
64	http://ulp.ub.ac.id	WordPress	3.5.1
65	http://vokasi.ub.ac.id	WordPress	3.8
66	http://poros.ub.ac.id	WordPress	4.3.1
67	http://display.ub.ac.id	WordPress	4.4
68	http://krismatiik.ub.ac.id/	WordPress	3.8.1
69	http://optiik.ub.ac.id	WordPress	4.0.1
70	http://thp.ub.ac.id	WordPress	4.5.2
71	http://aec.fia.ub.ac.id	WordPress	4.3.5
72	http://apatte62.ub.ac.id	WordPress	4.5.2
73	http://teknikkimia.ub.ac.id	WordPress	4.0

4.1.3 Pengumpulan Informasi Celah Keamanan

Berdasarkan hasil pendeteksian jenis website.Website di Universitas Brawijaya 80% menggunakan CMS WordPress dan sisanya menggunakan PHP manual / framework.Dari hasil ini,penulis kemudian mencari informasi terkait celah keamanan apa saja yang terdapat pada CMS WordPress dan juga pada PHP manual / framework.Untuk website yang menggunakan CMS WordPress, semua celah keamanan yang pernah di publikasikan tersimpan pada WPScan Vulnerability Database (WPVULNDB),terdapat ratusan celah keamanan mulai dari versi WordPress yang digunakan, template dan plugin pada WordPress dimana plugin ini dapat dikembangkan oleh pihak ketiga sehingga pada sistem yang akan dikembangkan akan memiliki fitur untuk mendeteksi celah keamanan berdasarkan data pada WPVULNDB, sedangkan untuk website yang menggunkan PHP manual / framework, celah keamanan yang sering terjadi adalah SQL Injection, Cross Site Scripting, Local File Include dan celah umum lainnya (OWASP,2013) sehingga khusus untuk website yang menggunakan PHP manual / framework pada sistem nantinya akan diimplementasikan fungsi untuk mendeteksi celah-celah keamanan yang umum berdasarkan data OWASP.

Penulis kemudian melakukan penelusuran lebih lanjut dan mendapatkan informasi bahwa terdapat tool yang berfungsi untuk mendeteksi celah keamanan pada CMS WordPress secara otomatis berdasarkan data pada WPVulnDB, yaitu tool WPScan (WordPress Vulnerability Scanner). WPScan menggunakan database *offline* yang bersumber dari WPVULNDB dimana jika terdapat pembaharuan (update) pada WPVULNDB maka pada database *offline* WPScan juga diperbaharui. Pada sistem yang dikembangakan, tool WPScan digunakan untuk proses deteksi celah keamanan khususnya pada website yang menggunakan WordPress. Sedangkan pada website yang menggunakan PHP Manual.Framework terdapat tool yang berfungsi untuk mencari celah keamanan yang umum, yaitu tool Uniscan, *tool* ini memiliki kelebihan, yaitu pada fitur *URL Crawling* dimana uniscan akan mengumpulkan semua URL pada website dan selanjutnya URL yang

dikumpulkan akan digunakan sebagai target *scanning* untuk mencari celah keamanan SQL Injection, Cross Site Scripting (XSS) dan Local File Include.

4.2 Lingkungan Penelitian

Lingkungan penelitian dirancang berdasarkan arsitektur jaringan di Universitas Brawijaya khususnya dalam manajemen website. Pada tahap ini, penulis melakukan observasi lebih lanjut mengenai arsitektur jaringan terhadap data-data domain website yang telah dikumpulkan.

Langkah pertama yang dilakukan dalam observasi ini adalah mencari informasi IP *address* Universitas Brawijaya yang bisa didapatkan melalui WHOIS. Ditemukan bahwa domain ub.ac.id menggunakan IP 175.45.184.70. IP ini kemudian digunakan untuk mencari informasi lebih detail menggunakan WHOIS, disini penulis menggunakan WHOIS dari APNIC (Asia-Pacific Network Information Centre) yang menyimpan database informasi IP address di seluruh Asia Pacific.

```
inetnum:        175.45.184.0 - 175.45.191.255
netname:        UNIBRAW-ID
descr:          Universitas Brawijaya
descr:          University / Direct Member IDNIC
descr:          Jl. Veteran
descr:          Malang, Jawa Timur, 65145
country:        ID
admin-c:        RW32-AP
tech-c:         RT155-AP
status:         ALLOCATED PORTABLE
remarks:        Send Spam & Abuse Reports to ratno@brawijaya.ac.id
mnt-by:         MNT-APJII-ID
mnt-irt:        IRT-IDNIC-ID
mnt-routes:     MAINT-ID-UNIBRAW
mnt-lower:      MAINT-ID-UNIBRAW
changed:        hm-changed@apnic.net 20091223
changed:        hm-changed@apnic.net 20151202
source:         APNIC
```

Gambar 4.2 WHOIS APNIC IP Universitas Brawijaya

Berdasarkan **Gambar 4.2**, hasil pencarian informasi IP melalui APNIC ditemukan bahwa Universitas Brawijaya menggunakan IP 175.45.184.0 – 175.45.191.255. Dari rentang IP yang digunakan ini, penulis melakukan penelurusan lebih lanjut mengenai IP address yang digunakan oleh masing-masing website yang telah dikumpulkan. Tabel 4.4 merupakan hasil pemetaan IP *Address* yang berhubungan dengan domain yang telah dikumpulkan.

Tabel 4.5 Hasil pemetaan IP Address

Domain Server	IP Server	Host
http://feb.ub.ac.id	175.45.187.180	http://elearningfeb.ub.ac.id
http://fp.ub.ac.id	175.45.185.194	http://iorc.fp.ub.ac.id

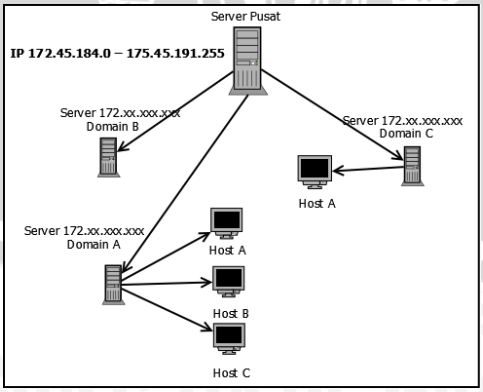
		http://tanah.ub.ac.id
		http://karyailmiah.fp.ub.ac.id
		http://sosek.ub.ac.id
http://hosting.ub.ac.id	175.45.184.160	http://apcaf.ub.ac.id
		http://industri.ub.ac.id
		http://io.ub.ac.id
		http://thp.ub.ac.id
		http://guesthouse.ub.ac.id
		http://bet.ub.ac.id
		http://icambbe.ub.ac.id
		http://aec.fia.ub.ac.id
		http://lab-srk.ub.ac.id
		http://tp.ub.ac.id
		http://lpke.ub.ac.id
		http://apatte62.ub.ac.id
		http://teknikkimia.ub.ac.id
		http://tip.ub.ac.id
		http://ppti.ub.ac.id
		http://ppsub.ub.ac.id
		http://spi.ub.ac.id
		http://pjm.ub.ac.id
		http://fisip.ub.ac.id
		http://optiik.ub.ac.id
		http://biosains.ub.ac.id
		http://bss.ub.ac.id
		http://sd.bss.ub.ac.id
		http://fapet.ub.ac.id
		http://griya.ub.ac.id
		http://icon.ub.ac.id
		http://kerjasama.ub.ac.id

		http://mku.ub.ac.id
		http://pemerintahan.ub.ac.id
		http://pkkss.ub.ac.id
		http://pilrek2014.ub.ac.id
		http://pemira.ub.ac.id
		http://ppikid.ub.ac.id
		http://ulp.ub.ac.id
		http://poros.ub.ac.id
		http://display.ub.ac.id
		http://krismatiik.ub.ac.id
		http://thp.ub.ac.id
		http://amd.ub.ac.id
		http://hmif.ub.ac.id
http://blog.ub.ac.id	175.45.184.5	-
http://kimia.ub.ac.id	175.45.184.87	http://bits.ub.ac.id
		http://fpik.ub.ac.id
		http://sipil.ub.ac.id
		http://lp3.ub.ac.id
		http://upkk.ub.ac.id
		http://em.ub.ac.id
		http://hukum.ub.ac.id
		http://upkk.ub.ac.id
-	114.4.66.126	http://smp.ub,ac.id
		http://fib.ub.ac.id
		http://isyg2015.ub.ac.id
		http://ppak.feb.ub.ac.id
		http://sikma.ub.ac.id
		http://bemtiik.ub.ac.id

-	175.45.185.188	http://eeccis.ub.ac.id
-	175.45.187.126	http://lib.ub.ac.id
http://filkom.ub.ac.id	175.45.187.242	-
http://psik.feb.ub.ac.id	175.45.187.179	http://accounting.feb.ub.ac.id
		http://bicab.ub.ac.id
		http://ie.feb.ub.ac.id
-	175.45.184.161	http://apis.ub.ac.id
		http://icoee2015.ub.ac.id
		http://pengairan.ub.ac.id
		http://pwk.ub.ac.id
		http://kepegawaian.ub.ac.id
		http://vokasi.ub.ac.id
		http://uaki.ub.ac.id
		http://arsitektur.ub.ac.id
		http://mansyla.ub.ac.id
		http://bak.ub.ac.id
		http://basicsc.ub.ac.id
		http://bakp.ub.ac.id
		http://soi2015.ub.ac.id
		http://mbesb.ub.ac.id
http://repository.mipa.ub.ac.id/	175.45.185.202	http://biologi.ub.ac.id
		http://mipa.ub.ac.id
-	175.45.187.74	http://fia.ub.ac.id

-	175.45.185.195	http://hme.ub.ac.id
		http://elektro.ub.ac.id
http://fk.ub.ac.id	175.45.187.196	http://pdg.fk.ub.ac.id
http://selma.ub.ac.id	175.45.184.30	-
http://singo.ub.ac.id	175.45.184.170	-
http://lecture.ub.ac.id	175.45.184.6	http://staff.ub.ac.id
http://teknik.ub.ac.id	175.45.185.220	-
http://rajabrawijaya.ub.ac.id	175.45.184.152	-
http://mesin.ub.ac.id	175.45.187.139	-

Berdasarkan tabel 4.5 didapatkan hasil bahwa dari rentang IP 175.45.184.0 – 175.45.191.255 menyimpan website-website yang digunakan oleh berbagai unit. Semua data domain yang dikumpulkan sesuai dengan rentang IP yang ditemukan pada APNIC. Dari hasil ini, penulis kemudian memiliki hipotesis mengenai arsitektur jaringan di Universitas Brawijaya khususnya dalam manajemen website dan sekaligus arsitektur ini digunakan sebagai lingkungan penelitian.Arsitektur ini dapat dilihat pada **Gambar 4.3**.



Gambar 4.3 Arsitektur Manajemen Website Universitas Brawijaya

Berdasarkan **Gambar 4.3**, terdapat server pusat yang mengatur rentang IP address dari 172.45.184.0 hingga 175.45.191.255.Ketika sebuah unit akan

mendaftarkan website,dari server ini kemudian ditentukan apakah akan diberikan IP *dedicated* ataukah akan disimpan pada server yang sudah ada atau *hosting*.

Perbedaan antara IP dedicated dan hosting adalah IP *dedicated* memiliki IP *address* khusus, tidak sama dengan IP address yang digunakan oleh website yang lainnya seperti terlihat pada **Gambar 4.3** dimana pada domain B yang memiliki IP address khusus sehingga jika kita mengakses IP address melalui browser, maka hasilnya akan sama seperti jika mengakses domain aslinya. Sedangkan *hosting* akan menggunakan web server yang sudah ada atau dapat mengguakan IP *dedicated* untuk membuat *virtual host* sehingga IP website yang menggunakan *hosting* memiliki IP *address* yang sama dengan IP *server* yang mengatur jalannya *web server*, contohnya pada **Gambar 4.3** terlihat domain A yang memiliki 3 virtual host dimana virtual host ini berupa website dan semua *host* memiliki IP *address* yang sama. Berbeda dengan IP *dedicated,* pada *hosting* ketika mengakses IP address melalui *browser* maka yang tampil pada *browser* adalah website yang mengatur jalannya *web server*,contohnya pada **Gambar 4.3** ketika mengakses IP address *host* A,B atau C maka yang akan tampil adalah website pada domain A.

4.3 Data Replikasi Website

Data replikasi website digunakan sebagai data pengujian. Data ini dibuat berdasarkan data domain yang telah dikumpulkan dengan melakukan validasi data-data terlebih dahulu. Adapun validasi data yang dibutuhkan adalah jenis website yang digunakan apakah menggunakan CMS atau PHP manual / framework yang sebelumnya sudah dibahas dimana dipilih pada website CMS dipilih CMS WordPress. Validasi selanjutnya adalah informasi versi dan plugin WordPress yang digunakan dimana CMS ini memiliki versi yang berbeda-beda dan juga plugin yang beragam.

Pembuatan data replikasi dalam penelitian ini dilakukan pada dua jenis website, yaitu WordPress dan PHP manual/framework. Data replikasi WordPress dibuatkan berdasarkan versi dan plugin yang digunakan pada data domain yang telah dikumpulkan sebelumnya, hanya akan dipilih website jenis WordPress dimana versi yag digunakan dapat terdeteksi.Sedangkan pada webite PHP manual/framework pembuatan datanya didapatkan dari tugas akhir Mahasiswa Fakultas Ilmu Komputer pada mata kuliah Pemrograman web atau Rekayasa Perangkat Lunak, juga diambil data website yang *vulnerable* melalui internet.

Pada pembahasan sebelumnya telah dibahas mengenai teknik dalam mendapatkan informasi versi dari WordPress, selanjutnya pada bagian ini dilakukan pencarian informasi pada plugin yang digunakan. Adapun teknik yang digunakan dalam mencari informasi plugin ini adalah dengan melakukan enumerasi. Enumerasi merupakan teknik penggalian informasi berdasarkan data yang telah ada, enumerasi plugin yang dilakukan dalam penelitian ini menggunakan tool WPScan dimana memiliki fitur *enumerate plugin* dimana fitur ini dapat mencari informasi plugin yang digunakan berdasarkan data-data yang terdapat pada website. Hasil enumerasi pada semua data dapat dilihat pada tabel-tabel di bawah ini.

Tabel 4.6 Data replikasi website

No	URL	Plugin
1	http://apcaf.ub.ac.id	akismet v2.6.0
		Contact-form v4.0.1
		Google-sitemap-generator v4.0.7.1
		Really-simple-capthca v1.8
		wordPress-seo v1.6.1
2	http://bet.ub.ac.id	akismet v2.5.9
		Contact-form 7 v3.6
		Jquery-t-countdown-widget v2.2.17
		wordPress-importer v0.6
		wordPress-language v1.1.1
		Wp-countdown-timer v1.0.0
		Wp-video-post v3.3
3	http://apis.ub.ac.id	akismet - v3.1.10
		easy-wp-smtp - v1.2.3
		profile-builder - v2.3.6
		wordpress-importer - v0.6.1
4	http://bak.ub.ac.id	akismet - v3.1.7
		easing-slider - v2.2.1.1
		event-organiser - v3.0.0
		ml-slider - v3.3.5
		nextgen-gallery - v2.1.23
5	http://arsitektur.ub.ac.id	content-slide - v1.4.2

		easing-slider - v2.1.2
		event-organiser - v2.3.2
		google-document-embedder - v2.5.22
		ml-slider - v3.3.1
		page-views-count - v1.3.2
		photo-gallery - v1.2.6
		qtranslate - v2.5.39
6	http://bakp.ub.ac.id	easy-spoiler - v1.9
		pretty-file-lister - v0.4
		shortcodes-ultimate - v4.9.9
		tinymce-advanced - v3.3.9.2
		wp-charts - v0.6.9.1
7	http://biologi.ub.ac.id	contact-form-7 - v4.3
		contact-form-plugin - v3.68
		content-slide - v1.4.2
		easy-media-gallery - v1.2.27
		gallery-plugin - v4.1.1
		google-sitemap-generator - v3.3
		page-links-to - v2.9.3
		qtranslate - v2.5.38
		sitemap - v4.2
8	http://bits.ub.ac.id	akismet - v2.5.9
		contact-form-7 - v2.4.6
		countdown-timer - v3.0.6

		easing-slider - v1.2.1
		event-organiser - v2.3.2
		jquery-t-countdown-widget - v2.2.16
		ml-slider - v3.3.1
		nextgen-gallery - v1.9.13
		qtranslate - v2.5.39
		wp-db-backup - v2.2.3
		wp-pagenavi - v2.31
		wp-recaptcha - v3.1.3
9	http://ppsub.ub.ac.id	blogger-importer - v0.9
		contact-form-plugin - v3.93
		fancybox-for-wordpress - v3.0.6
		movabletype-importer - v0.4
		multi-column-tag-map - v12.0.4
		photo-gallery - v1.2.71
		qtranslate - v2.5.32
		rss-importer - v0.1
		tumblr-importer - v0.8
		wordpress-importer - v0.6
		wp-paginate - v1.3.1
		wp-tooltip - v1.0.1
		wpcat2tag-importer - v0.5
10	http://ppti.ub.ac.id	akismet - v3.0.0
		contact-form-7 - v2.4.6
		easy-faq-with-expanding-text - v3.2.8.3.1

		form-maker - v1.7.94
		ninja-forms - v2.9.28
		our-team-enhanced - v2.0
		qtranslate - v2.5.39
		team-members - v1.3.1
		wp-db-backup - v2.2.3
		wp-recaptcha - v3.1.3
11	http://fapet.ub.ac.id	easing-slider - v2.1.3
		qtranslate - v2.5.34
12	http://teknikkimia.ub.ac.id	count-per-day - v3.2.9
		easing-slider - v2.1.3
		qtranslate - v2.5.35
		simple-sitemap - v1.52
13	http://fia.ub.ac.id	bogo - v2.8
		broken-link-checker - v1.11.2
		disable-comments - v1.5
		easy-wp-smtp - v1.2.3
		ewww-image-optimizer - v2.8.5
		image-widget - v4.2.2
		lightbox-gallery - v0.8.3
		ml-slider - v3.3.6
		newstatpress - v1.2.4
		nextgen-gallery - v2.1.44
		php-code-widget - v2.3
		simple-history - v2.7.3
		theme-check - v20160523.1

		wordpress-importer - v0.6
		wordpress-popup - v4.8.0.0
		wp-limit-login-attempts - v2.4.6
14	http://fisip.ub.ac.id	1000grad-epaper - v1.4.12
		content-slide - v1.4.2
		e-paper - v1.15
		qtranslate - v2.5.34
		sitemap - v4.2
		wp-statistics - v8.8
15	http://hukum.ub.ac.id	akismet - v2.5.9
		mce-table-buttons - v2.0
		ml-slider - v2.3-beta
		qtranslate - v2.5.39
		sitemap - v4.2
16	http://icambbe.ub.ac.id	akismet - v2.6.0
		all-in-one-seo-pack - v1.2.8
		blogger-importer - v0.7
		breadcrumb-navxt - v5.0.1
		easing-slider - v2.1.3
		google-language-translator - v2.6
		google-sitemap-generator - v3.2.8
		wordpress-importer - v0.6.1
17	http://icoee2015.ub.ac.id	contact-form-7 - v4.1.1

		event-espresso - v3.1.30
		force-regenerate-thumbnails - v2.0.3
		google-sitemap-generator - v4.0.7.1
		jetpack - v3.4.3
		wordpress-importer - v0.6.1
		wp-mail-smtp - v0.9.5
18	http://icon.ub.ac.id	akismet - v2.5.3
19	http://industri.ub.ac.id	akismet - v3.0.0
		contact-form-to-email - v1.0.1
		easy-fancybox - v1.5.7
		scroll-back-to-top - v1.1.3
		ultimate-posts-widget - v1.8.1
		wowslider - v5.
20	http://io.ub.ac.id	akismet - v3.0.0
		breadcrumb-navxt - v5.0.0
		easing-slider - v2.1.2
		easy-table - v1.5.2
		jetpack - v3.9.1
		tablepress - v1.6
		team - v1.5
		tinymce-advanced - v4.1.1
		ultimate-tables - v1.3
		wp-newsticker - v2.0
		wp-page-numbers - v0.5
		wp-team-manager - v4.3

21	http://upkk.ub.ac.id	akismet - v2.5.9
		easing-slider - v2.1.2
		event-organiser - v2.3.2
		gallery-plugin - v4.0.2
		jetpack - v4.0.2
		ml-slider - v3.3.1
		nextgen-gallery - v1.9.13
		qtranslate - v2.5.39
		velvet-blues-update-urls - v3.1
22	http://kepegawaian.ub.ac.id	akismet - v3.1.6
		event-organiser - v2.13.7
		iframe - v4.2
		jetpack - v3.8.1
		ml-slider - v3.3.5
		wp-file-upload - v3.6.1
		wp-newsticker - v2.0
23	http://kerjasama.ub.ac.id	fancy-gallery - v1.0
		ml-slider - v3.2
		qtranslate - v2.5.39
24	http://lab-srk.ub.ac.id	contact-form-7 - v4.0.1
		contact-form-to-email - v1.1.5
		cyclone-slider-2 - v2.8.3
		easy-table - v1.5.2
		google-language-translator - v4.0.5
		ml-slider - v3.3.1

		smooth-slider - v2.6.4
		tablepress - v1.4
		ultimate-tables - v1.6.3
		wp-file-upload - v3.8.4
		wp-maintenance-mode - v2.0.3
25	http://lpke.ub.ac.id	easing-slider - v2.1.4.3
26	http://mipa.ub.ac.id	akismet - v2.5.9
		contact-form-7 - v3.5.4
		count-per-day - v3.2.9
		easing-slider - v2.1.2
		google-sitemap-generator - v3.3
		nextgen-gallery - v1.9.13
		page-links-to - v2.9.3
		php-code-for-posts - v1.1.3
		qtranslate - v2.5.38
		qtranslate-slug - v1.1.7
		sitemap - v4.2
		ultimate-posts-widget - v1.7
27	http://pdg.fk.ub.ac.id	all-in-one-seo-pack - v2.2.7
		better-wp-security - v3.5.5
		contact-form-7 - v3.5.2
		easing-slider - v2.1.4.3
		lockdown-wp-admin - v2.1
		page-links-to - v2.9.4
		qtranslate - v2.5.35

		sitemap - v4.3
28	http://pengairan.ub.ac.id	acf-qtranslate - v1.7.9
		akismet - v3.1.7
		auto-post-thumbnail - v3.4.0
		breadcrumb-navxt - v5.0.0
		contact-form-plugin - v3.84
		easing-slider - v2.1.2
		google-document-embedder - v2.6
		jetpack - v3.2.1
		ml-slider - v3.3.5
		page-links-to - v2.9.3
		page-views-count - v1.3.2
		qtranslate - v2.5.39
29	http://pilrek2014.ub.ac.id	akismet - v2.5.9
		breadcrumb-navxt - v5.0.0
		easing-slider - v2.1.2
		events-manager - v5.5.2
		ga-google-analytics - v20140123
		tinymce-advanced - v3.5.9
30	http://pjm.ub.ac.id	cardoza-wordpress-poll - v35.2
		easing-slider - v1.2
		easy-spoiler - v1.9
		google-language-translator - v5.0.05
		jetpack - v2.7

		promotion-slider - v3.3.4
		nextgen-scrollgallery - v1.8
		qtranslate - v2.5.39
		stats-counter - v1.2.2.5
		tinymce-advanced - v3.5.8
		wordpress-importer - v0.6
31	http://ppikid.ub.ac.id	contact-form-7 - v3.9.3
		contact-form-plugin - v3.83
		easing-slider - v1.2
		embed-any-document - v2.1
		google-captcha - v1.07
		universal-post-manager - v1.4.1
32	http://pwk.ub.ac.id	akismet - v3.0.4
		ml-slider - v3.2
		qtranslate - v2.5.39
		regenerate-thumbnails - v2.2.4
		youtube-video-player - v1.0.3
33	http://selma.ub.ac.id	akismet - v2.5.3
		ml-slider - v3.3.6
		qtranslate - v2.5.20
		qtranslate-x - v3.4.6.4
		w3-total-cache - v0.9.4
34	http://singo.ub.ac.id	akismet - v2.5.9
		qtranslate - v2.5.39

		wp-db-backup - v2.2.4
		wp-recaptcha - v3.2
35	http://sipil.ub.ac.id/	akismet - v3.1.1
		backupwordpress - v2.6.2
		contact-form-7 - v4.0.1
		easing-slider - v2.1.2
		event-organiser - v2.3.2
		flamingo - v1.2
		google-language-translator - v4.0.5
		ml-slider - v3.0
		nextgen-gallery - v1.9.13
		wp-sitemap-page - v1.3.0
36	http://staff.ub.ac.id	buddypress - v2.0.2
		disable-comments - v1.3.1
		feedwordpress - v2014.0805
		google-analytics-for-wordpress - v4.3.5
		hyper-cache - v3.0.2
		tinymce-advanced - v4.0.1
		wordpress-importer - v0.6
		wp-spamfree - v2.1.1.2
37	http://spi.ub.ac.id	easing-slider - v2.1.4.3
		easy-table - v1.6
		nextgen-gallery - v2.1.16
		qtranslate - v2.5.39
		wp-team-manager - v4.5

38	http://tanah.ub.ac.id	all-in-one-seo-pack - v2.3.4.2
		jetpack - v4.0.2
		ml-slider - v3.3.5
		pdf-print - v1.8.7
		qtranslate-x - v3.4.6.4
		sitemap - v4.3
		tinymce-advanced - v4.2.5
39	http://teknik.ub.ac.id	akismet - v2.5.9
		easing-slider - v1.2.1
		embed-any-document - v2.2.1
		gotmls - v4.16.17
		google-sitemap-generator - v4.0.7.1
		mce-table-buttons - v3.2
		ml-slider - v2.8-beta
		nextgen-gallery - v2.0.66.27
		page-views-count - v1.3.1
		sitemap - v4.2
		ultimate-posts-widget - v1.6
40	http://tip.ub.ac.id	akismet - v2.5.6
		bootstrap-for-contact-form-7 - v1.3.0
		contact-form-7 - v4.3.1
		qtranslate-x - v3.4.6.4
		really-simple-captcha - v1.8.0.1
		sitemap - v4.2

		wp-stats - v2.50
41	http://uaki.ub.ac.id	akismet - v3.1.7
		contact-form-7 - v4.3.1
		digg-digg - v5.3.6
		easy-table - v1.5.3
		mini-twitter-feed - v2.0.1
		post-types-order - v1.8.5
		rotatingtweets - v1.8.1
		visitor-maps - v1.5.8.10
		wordpress-importer - v0.6
		wp-statistics - v10.0.1
		wp-table-reloaded - v1.9.4
42	http://ulp.ub.ac.id	dynamic-featured-image - v3.3.0
		header-image-slider - v0.3
		qtranslate - v2.5.34
		wptouch - v3.6.5
43	http://vokasi.ub.ac.id	akismet - v3.1.7
		slider-image - v3.1.87
44	http://poros.ub.ac.id	contact-form-7 - v4.3.1
		iframe - v4.2
		maintenance - v2.6
		page-scroll-to-id - v1.6.0
		regenerate-thumbnails - v2.2.6
		social-media-feather - v1.7.9
		twitter-widget-pro - v2.8.0

		widget-settings-importexport - v1.5.0
		woocommerce - v2.4.13
		woosidebars - v1.4.3
		wordpress-importer - v0.6.1
		wp-statistics - v9.7
45	http://display.ub.ac.id	contact-form-7 - v4.4
		disqus-comment-system - v2.84
		google-analyticator - v6.4.9.7
		jetpack - v3.9.7
		maintenance - v2.7.1
		the-events-calendar - v4.1.0.1
		tiled-gallery-carousel-without-jetpack - v2.1
		visual-form-builder - v2.8.6
		wp-image-zoooom - v1.2.6
		yith-maintenance-mode - v1.3.1
		youtube-embed-plus - v11.0.1
46	http://krismatiik.ub.ac.id/	akismet - v2.5.9
		gallery-plugin - v4.1.3
		google-sitemap-generator - v3.3
		kebo-twitter-feed - v1.4.4
		php-code-for-posts - v1.2.0
		social - v2.11

		the-events-calendar - v3.4.1
		visual-form-builder - v2.8
		wd-twitter-feed - v1.2.1
		widget-settings-importexport - v1.2
		wordpress-importer - v0.6.1
47	http://thp.ub.ac.id	breadcrumb-navxt - v5.4.0
		contact-form-7 - v4.4.2
		ppqtranslate - v2.7.2
		recent-tweets-widget - v1.6.5
		woosidebars - v1.4.3
48	http://optiik.ub.ac.id	cleantalk-spam-protect - v4.18
		contact-form-7 - v4.1
		count-per-day - v3.4
		ct-twitter - v3.4
		dynamic-to-top - v3.4.2
		jetpack - v3.3.2
		option-tree - v2.5.1
		options-framework - v1.8.3
		page-links-to - v2.9.4
		responsive-lightbox - v1.4.10
		si-captcha-for-wordpress - v2.7.7.5
		twitter-js - v0.2
		wordpress-importer - v0.6
		wordpress-seo - v1.7.3.1

		wp-maintenance-mode - v2.0.3
49	http://aec.fia.ub.ac.id	contact-form-7 - v4.3.1
		contact-form-7-to-database-extension - v2.9.12
		google-document-embedder - v2.5.22
		wptouch - v3.6.6
50	http://apatte62.ub.ac.id	Ultimate_VC_Addons
		contact-form-7 - v4.4.2
		gallery-plugin - v4.2.9
		recent-tweets-widget - v1.6.5
		woocommerce - v2.5.5
51	http://em.ub.ac.id	Tidak terdeteksi
52	http://fpik.ub.ac.id	Tidak terdeteksi
53	http://bicab.ub.ac.id	akismet - v2.5.6
		contact-form-7 - v3.3.3
		really-simple-captcha - v1.5
		simple-google-map - v2.0
		sociable - v4.3.3
		wordpress-importer - v0.6
		wp-pagenavi - v2.73
54	http://eeccis.ub.ac.id/2016/	countdown-timer - v3.0.6
		events-manager - v5.6.2

55	http://tp.ub.ac.id	akismet - v2.5.9
		ml-slider - v3.3.4.1
		qtranslate - v2.5.38
		wp-parsi-statistics - v1.5
56	http://kimia.ub.ac.id	akismet - v3.1.5
		contact-form-7 - v3.4.2
		google-sitemap-generator - v4.0.7.1
		ml-slider - v3.3.7
		page-links-to - v2.9.4
		qtranslate - v2.5.39
		simple-sitemap - v1.65
		sitemap - v4.2
		tinymce-advanced - v4.1.1

Data pengujian khusus untuk website jenis WordPress dibuat berdasarkan informasi pada **Tabel 4.5** dimana menyesuaikan versi dan plugin yang digunakan.Dari hasil pendeteksian plugin, terdapat 2 website yang tidak dapat terdeteksi sehingga total data pengujian yang akan digunakan pada website jenis WordPress berjumlah 54 website. Sedangkan data pengujian pada website PHP manual / framework yang didapatkan dari Mahasiswa Fakultas Ilmu Komputer dan website celah keamanan yang bersumber dari internet dapat dilihat pada tabel 4.6, data ini sekaligus dapat digunakan tanpa ada perubahan kode dari sumbernya.

Tabel 4.7 Data pengujian website PHP manual/framework

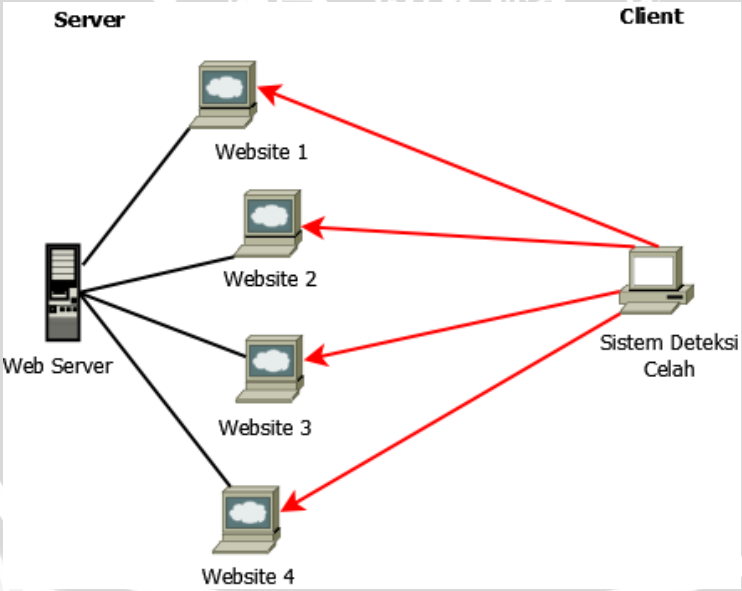
No	Sumber	URL	Keterangan
1	Mohammad Vicky Agassi (125150207111004)	http://localhost/mtic	Tugas akhir RPL
2	Tatag Winaryo (125150207111009)	http://localhost/nat	Tugas akhir RPL
3	Abdullah (125150207111002)	http://localhost/sete	Data Latih

4	Sukma Wardana Hadi P (125150207111007)	http://localhost/infokos/public	Tugas Akhir Pemweb
5	EVA-00 (eva-00.web.id)	http://localhost/injector	Web simulasi bug
6	Azwar Mustofa Wahyudi (125150207111013)	http://localhost/mtq3	Tugas Akhir Pemrograman Web

Dalam penelitian ini, website yang dapat direplikasi hanya website yang menggunakan CMS dimana berdasarkan data yang ditemukan menggunakan CMS WordPress,alasan dapat dilakukannya replikasi data dikarenakan data yang didapatkan dapat direplikasi sesuai dengan data asli, sedangkan untuk PHP manual/framework tidak dapat dilakukan replikasi dikarenakan tidak adanya metode yang sesuai untuk melakukan validasi data.

4.4 Lingkungan Pengujian

Pengujian sistem dilakukan pada semua data uji yang telah direplikasi. Dalam pengujian ini, penulis membuat sebuah web *server* yang menyimpan website-website dalam satu server (hosting) yang sebelumnya telah dijelaskan pada linkgungan penelitian.Lingkungan pengujian secara keseluruhan dapat dilihat pada **Gambar 4.4**.



Gambar 4.4 Lingkungan pengujian

Berdasarkan **Gambar 4.4**, satu web *server* menyediakan semua data replikasi yang telah dibuat baik itu WordPress maupun PHP manual/framework, *server* beserta website dibuat secara *offline* pada komputer lokal peneliti. Kemudian sistem deteksi juga ditempatkan pada komputer lokal peneliti, sehingga dalam

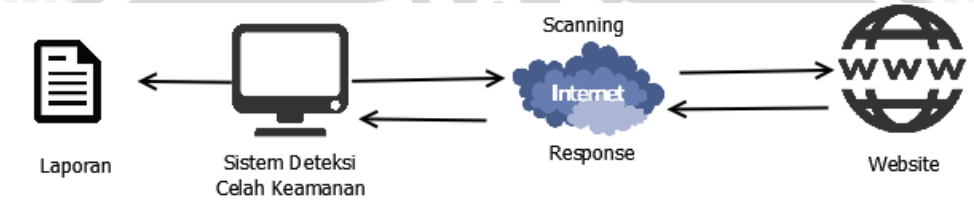
pengujian ini hanya menggunakan satu laptop yang berisi semua data replikasi beserta sistem deteksi celah keamanan.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada Bab 5 ini membahas mengenai perancangan sistem yang akan dibagun beserta implementasinya. Akan dijelaskan gambaran dan alur kerja sistem secara umum dilanjutnya dengan perancangan secara detail dan selanjutya masuk pada tahap implementasi.

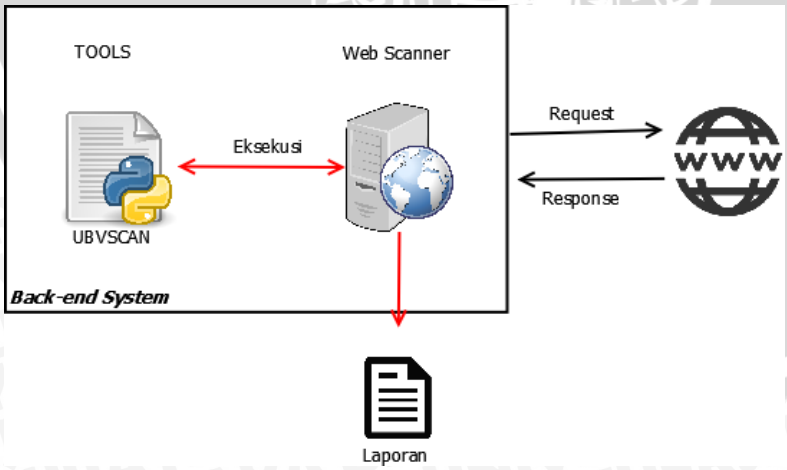
5.1 Gambaran Umum Sistem

Gambaran umum sistem menjelaskan secara umum bagaimana sistem deteksi celah keamanan bekerja dimana akan dijelaskan dari dua sisi, yaitu *front-end* dan *back-end*.



Gambar 5.1 *Front-end* Sistem

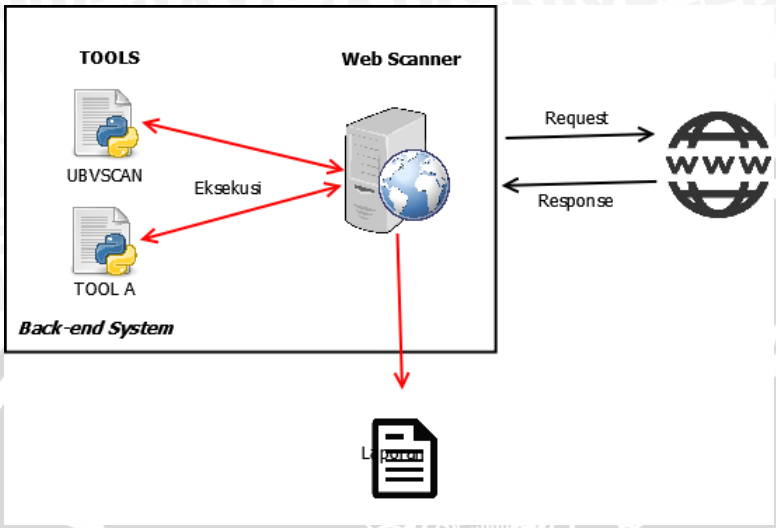
Pada sisi *front-end* dapat dilihat pada **Gambar 5.1** dimana langkah awal yang dilakukan oleh sistem adalah melakukan *ping testing* untuk mendeteksi apakah web sedang aktif atau tidak, jika web tidak aktif maka sistem akan menghentikan proses untuk *scanning*, jika web terdeteksi aktif maka selanjutnya sistem akan menjalankan *engine* utama yang bernama UBVSCAN (UB Vulnerability Scanner). UBVSCAN akan melakukan deteksi jenis website yang digunakan dan selanjutnya melakukan scanning berdasarkan jenis website untuk mencari celah keamanan hingga proses selesai dan hasil akhirnya akan dibuat laporan yang tersimpan pada sistem.



Gambar 5.2 *Back-end* sistem

Pada sisi *back-end* seperti terlihat pada **Gambar 5.2**, sistem ini berbasis website *scanner* yang dapat diakses via browser,website ini digunakan sebagai perantara untuk mengeksekusi *engine* atau *tool* utama dari sistem, yaitu UBVSCAN

yang dalam hal ini dibuat dengan menggunakan pemrograman Python, kemudian hasil eksekusi akan ditampilkan pada *front-end*. Website scanner tidak hanya digunakan sebagai perantara untuk *engine* utama namun dapat digunakan untuk semua *tools* yang dapat ditambahkah oleh administrator seperti terlihat pada **Gambar 5.3**.



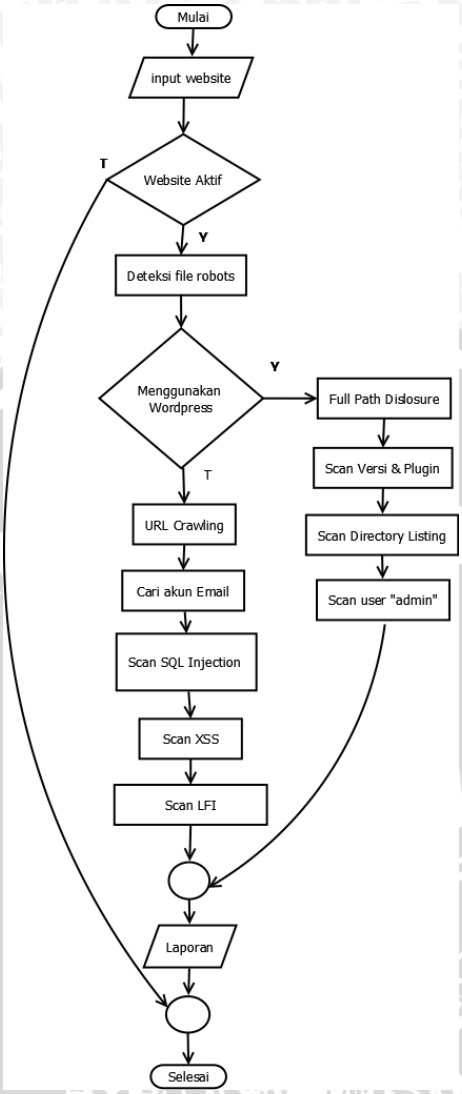
Gambar 5.3 Penambahan *Tools*

5.2 Alur Kerja Sistem

Berdasarkan data penelitian yang telah dikumpulkan, proses pembuatan sistem akan berfokus untuk melakukan deteksi celah keamanan pada website jenis WordPress dan PHP manual / framework. Adapun alur kerja sistem deteksi celah keamanan secara keseluruhan yang akan dikembangkan dapat dilihat pada **Gambar 5.4**.

Berdasarkan **Gambar 5.4**, perbedaan pada dua jenis website membuat peneliti membangun sistem dimana memilki mekanisme scanning yang berbeda.Sistem yang dikembangkan harus mampu melakukan deteksi celah keamanan berdasarkan jenis website. Setelah berhasil melakukan deteksi jenis website maka sistem dapat menentukan mekanisme scanning apakah akan melakukan *scanning* pada website jenis WordPress ataukan pada website PHP manual / framework.Untuk website yang menggunakan WordPress maka sistem harus dapat melakukan deteksi terhadap celah keamanan sebagai berikut :

- File Robots
- Full Path Disclosure
- Celah Keamanan pada versi dan plugin WordPress
- Username default ‘admin’
- Directory listing



Gambar 5.4 Flowchart Alur Kerja Sistem

Sedangkan untuk website yang menggunakan PHP manual / framework,maka sistem yang dibangun harus dapat melakukan deteksi pada celah keamanan sebagai berikut :

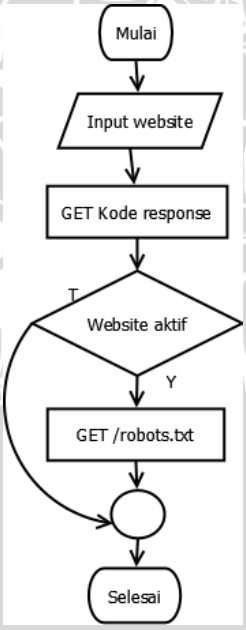
- File Robots
- Pencarian akun *e-mail*
- SQL Injection
- Cross Site Scripting
- Local File Inclusion

5.3 Perancangan Sistem Deteksi Celah Keamanan

5.3.1 Tahapan Perancangan

Tahapan perancangan merupakan langkah-langkah yang dilakukan oleh peneliti dalam melukan perancangan sistem deteksi celah keamanan. Adapun kebutuhan yang diperlukan pada perancangan sistem adalah data pendukung yang telah dibahas pada BAB 4. Data yang telah didapatkan pada BAB 4 menjadi referensi dalam mengembangkan sistem yang berfokus untuk mencari celah keamanan pada website CMS WordPress dan PHP manual/framework.

Pada **Gambar 5.4** dapat dilihat bahwa ketika pengguna memasukkan URL, sistem terlebih dahulu melakukan deteksi apakah website aktif atau tidak. Proses untuk melakukan deteksi apakah website aktif atau tidak adalah dengan cara melihat kode *response* dari website target. Jika kode yang dikembalikan dari target adalah 200, maka website dinyatakan aktif, namun jika tidak mendapatkan respon dari server maka server dinyatakan tidak aktif dan proses scanning selesai namun jika website aktif maka sistem langsung menjalankan deteksi file *robots*. Metode deteksi ini merupakan tahapan kedua ketika sistem telah dijalankan, namun dalam pendeteksian celah keamanan, deteksi file robots merukapan deteksi celah keamanan yang pertama yang digunakan untuk semua jenis website. Kedua metode ini, yaitu deteksi website aktif dan file robots digabungkan dalam satu metode dalam program.Alur kedua metode ini **Gambar 5.5.**



Gambar 5.5 Alur Deteksi Website Aktif dan File Robots

Setelah melakukan deteksi pada file robots, sistem selanjutnya melakukan deteksi jenis website, sistem melakukan deteksi pada dua jenis website, yaitu WordPress dan PHP manual/framework. Adapun teknik yang digunakan dalam melakukan deteksi jenis website dibahas pada sub-bab selanjutnya. Namun yang perlu dijelaskan lebih lanjut adalah proses deteksi celah keamanan pada kedua jenis website. Adapun langkah-langkah yang dilakukan oleh sistem ketika

melakukan deteksi celah keamanan pada website yang menggunakan WordPress adalah sebagai berikut.

1. Full Path Disclosure, pada teknik ini sistem mencari *path* atau direktori dimana ketika diakses mengandung pesan *error* sehingga menampilkan path secara lengkap misalnya /var/ww/html/index.php.
2. Deteksi celah keamanan pada versi WordPress, pada deteksi ini, sistem melakukan deteksi versi WordPress yang digunakan, kemudian dari versi yang ditemukan dicocokkan dengan database celah keamanan berbentuk file json yang bersumber dari WPVULNDB. Jika cocok maka sistem menampilkan daftar celah keamanan sesuai dengan versi yang digunakan.
3. Deteksi celah keamanan pada plugin WordPress, pada deteksi ini tidak jauh berbeda dengan deteksi celah keamanan pada versi WordPress. Sistem melakukan deteksi plugin yang digunakan beserta versinya dan kemudian akan mencocokkan hasilnya dengan database celah keamanan plugin berbentuk file json yang bersumber dari WPVULNDB.
4. Deteksi directory listing, pada deteksi ini, sistem melakukan melakukaan pengecekan pada database apakah pada versi WordPress atau plugin WordPress terdapat *directory listing*.
5. Deteksi username admin, pada deteksi ini dilakukan metode untuk mendeteksi username default, yaitu admin. Cara yang digunakan ada dua, jika salah satu cara yang digunakan berhasil maka sistem akan menampilkan bahwa terdapat username admin. Berikut adalah dua metode untuk mendeteksi username WordPress :
    - Mengakses URL dengan format http:/[web]/author?=1
    - Mengakses URL dengan format http:/[web]/author/admin

Metode-metode diatas diterapkan dalam sistem khusus untuk melakukan deteksi celah keamanan pada WordPress, sedangkan metode deteksi celah keamanan untuk PHP manual/framework adalah sebagai berikut:

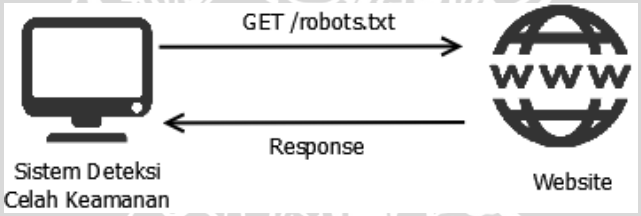
1. Mencari akun email, dalam deteksi ini sistem melakukan pencarian email dengan melakukan parsing data HTML dan mencari email dengan format [*]@[*].[*], contohnya email_tes@gmail.com.
2. Deteksi SQL Injection, deteksi SQL injection dilakukan dengan cara memasukkan tanda petik (‘) pada setiap parameter yang ditemukan, jika muncul pesan error, sistem akan menampilkan bahwa website terdeteksi SQL Injection, jika tidak ada pesan error maka celah tidak ditemukan.
3. Deteksi Cross Site Scripting (XSS), deteksi XSS hampir sama dengan pendeteksian celah SQL Injection dimana memasukkan *payload* pada parameter yang ditemukan, namun *payload* yang digunakan adalah kode-kode *javascript* untuk menampilkan celah XSS tersebut. Contoh *payload* yang digunakan misalnya *<script>alert(‘XSS)</script>.*

4. Deteksi Local File Include, deteksi ini sama dengan pendeteksian SQL injection dan XSS, hanya perbedaannya terletak pada *payload* yang digunakan, jika pada SQL injection menggunakan tanda petik satu dan pada XSS menggunakan kode javascript, pada Local File Include,payload yang digunakan adalah perintah (command) berbahaya yang dapat dijalankan pada sistem misalnya perintah untuk membaca konfigurasi pengguna linux, yaitu *cat /etc/passwd*.

Pada dasarnya deteksi celah keamanan yang digunakan pada PHP manual/framework memiliki metode yang sama, yaitu mencari URL yang memiliki parameter misalnya parameter *id=1* yang terdapat pada URL berikut http://[web]/index.php?id=1. Pada URL tersebut kemudian bisa dimasukkan berbagai macam *payload* SQL injection, XSS dan Local File Include.

5.3.2 Perancangan Deteksi File Robots

Perancangan ini berhubungan dengan teknik *information gathering* pada *penetration testing*. File robots berfungsi untuk menghindari file dan direktori yang dimiliki dari *crawling search engine* contohnya Google yang melakukan crawling pada setiap website yang ditemui dan melakukan index pada setiap file dan direktori yang ditemukan. Alasan dibuatnya deteksi ini adanya kemungkinan dalam file robots terdapat file atau direktori sensitif yang belum diketahui oleh user maupun search engine. Adapun fitur deteksi file robots dapat dilihat pada **Gambar 5.6**.

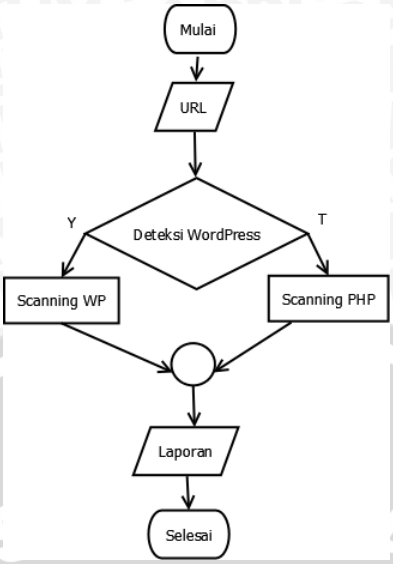


Gambar 5.6 Deteksi File Robots

Berdasarkan **Gambar 5.6**, Pada sistem yang dibangun, setelah website berhasil mendeteksi bahwa website aktif, maka sistem akan melakukan request file robots dengan perintah “GET /robots.txt”. Jika file ditemukan maka sistem akan mengakses file tersebut dan kemudian menampilkan hasilnya pada laporan hasil *scanning*.

5.3.3 Perancangan Deteksi Jenis Website

Pendeteksian jenis website merupakah hal terpenting dalam sistem. Pada engine utama ini, pendeteksian jenis website menentukan mekanisme scanning pada jenis website.Adapun alur kerja deteksi jenis website dapat dilihat pada **Gambar 5.7.**



Gambar 5.7 Flowchart Deteksi Jenis Website

Berdasarkan gambar diatas, secara spesifik pada UBVSCAN terdapat dua metode deteksi, yaitu metode deteksi website jenis WordPress dan deteksi PHP manual / framework. Pendeteksian WordPress dilakukan dengan mendeteksi file dan direktori default, yaitu:

- /wp-content
- /wp-login.php

Pendeteksian ini dilakukan pertama kali pada saat sistem dijalankan, jika direktori wp-content tidak ditemukan, maka sistem akan mencoba untuk mendeteksi file wp-login.php sehingga pada implementasinya akan digunakan perintah sebagai berikut:

- GET /wp-content atau
- GET /wp-login.php

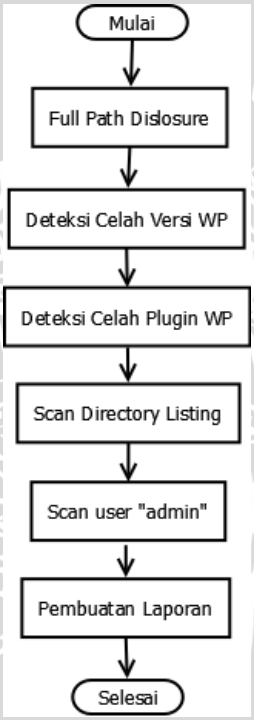
Untuk membuktikan apakah file dan direktori tersedia maka server akan memberikan status code 200 OK sedangkan jika tidak tersedia maka server akan memberikan respon 404 Not Found.

Sistem akan mendeteksi website jenis WordPress terlebih dahulu, jika file dan direktori yang diminta ditemukan maka sistem akan menyimpulkan website tersebut adalah WordPress dan selanjuntnya menjalankan mekanisme scanning WordPress dengan menggunakan *tool* WPScan yang telah dimofikasi pada UBVSCAN. Jika sistem tidak menemukan file dan direktori yang diminta maka sistem akan menganggap website tersebut menggunakan PHP manual / framework dan selanjutnya akan menjalankan mekanisme scanning PHP manual / framework dengan memanfaatkan *tool* uniscan yang juga sudah dimodifikasi pada UBVSCAN.

5.3.4 Perancangan Mekanisme Scanning WordPress

Seperti yang telah dibahas sebelumnya, ketika sistem melakukan deteksi jenis website selanjutnya sistem akan memilih mekanisme scanning dalam hal ini yang akan dibahas adalah mekanisme scanning pada website jenis WordPress. Mekanisme ini memanfaatkan tool WPScan. WPScan digunakan karena pencarian celah keamanan berdasarkan WPVULNDB. Dalam penggunaanya pada *engine* utama hasil dari wpscan diambil informasi yang penting saja. Penulis membuat script dari Python yang berfungsi untuk melakukan *parsing data* pada output WPScan. Berikut adalah data yang di-*parsing*:

- Full Path Disclosure
- Celah Keamanan pada versi dan plugin WordPress
- Username default 'admin'
- Directory listing



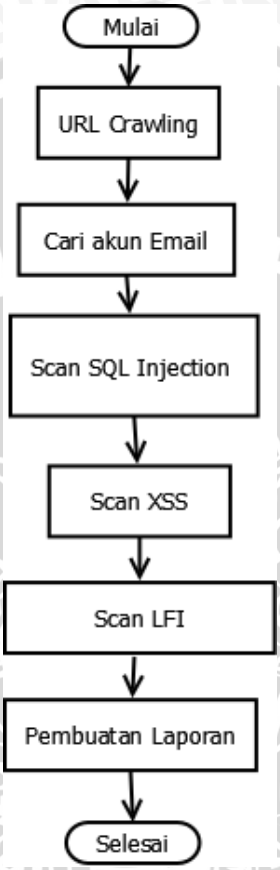
Gambar 5.8 Alur Deteksi Celah Keamanan Pada WordPress

Semua data ini ditampilkan pada laporan *scanning*. Khusus untuk celah keamanan pada versi dan plugin WordPress akan ditambahkan link menuju website WPVULNDB dengan tujuan agar user mengetahui informasi detail serta solusi untuk memperbaiki celah keamanan tersebut. Adapun alur deteksi celah keamanan pada WordPress dapat dilihat pada **Gambar 5.8.**

5.3.5 Perancangan Mekanisme Scanning PHP manual /framework

Mekanisme scanning PHP manual / framework memanfaatkan tool Uniscan yang akan dimodifikasi pada *engine* utama. Alasan penulis memilih tool Uniscan

adalah terdapat fitur *URL Crawling* dimana semua URL yang terdapat pada website akan dikumpulkan dan hasil pengumpulan URL akan digunakan sebagai target scanning sehingga proses untuk mencari celah keamanan dapat dilakukan secara menyeluruh. Celah keamanan yang ditelusuri oleh Uniscan berfokus pada input dari pengguna yang dapat membuat pesan error. Adapun alur deteksi celah keamanan pada website PHP manual/framework dapat dilihat pada **Gambar 5.9**.



Gambar 5.9 Alur Deteksi Celah Keamanan Website PHP manual/framework

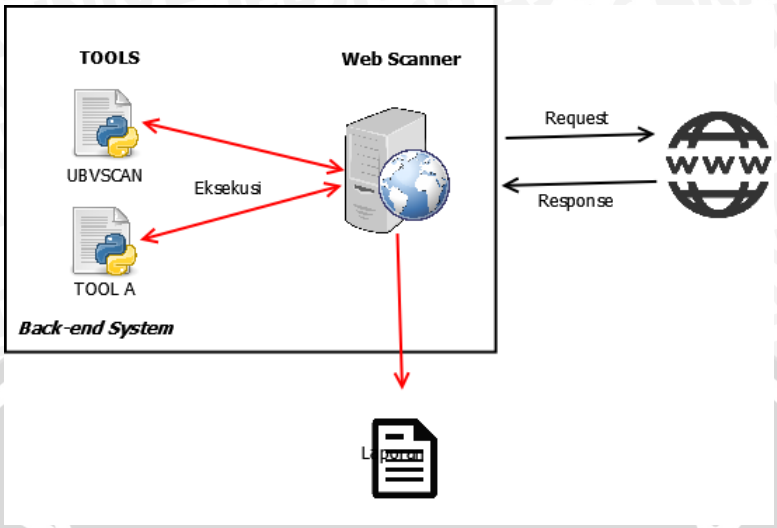
Dalam penggunaannya pada *engine* utama, dilakukan *parsing* data untuk mengambil informasi penting meliputi:

- Email yang terdapat pada website
- SQL Injection
- Cross Site Scripting
- Local File Inclusion

5.3.6 Perancangan Website

Website dalam penelitian ini berfungsi sebagai perantara yang berfungsi untuk mengeksekusi *script* Python melalui website sehingga hasil eksekusi script juga dapat dilihat langsung melalui website. Selain berfungsi untuk eksekusi *script*, website ini juga digunakan dalam membuat laporan hasil *scanning* yang diambil

dari hasil eksekusi *script*. Proses eksekusi *script* dan proses pembuatan laporan yang dilakukan oleh website dapat dilihat pada **Gambar 5.10**.



Gambar 5.10 Proses Eksekusi Script dan Pembuatan Laporan

Seperti terlihat pada **Gambar 5.10**, website hanya digunakan sebagai jembatan agar UBVSCAN (*script* Python) dapat dieksekusi dan menampilkan hasilnya website tersebut dalam bentuk laporan hasil scanning. Adapun format laporan yang ditentukan dalam penelitian ini terbagai menjadi dua, yaitu untuk website yang menggunakan WordPress dan PHP manual/framework, format ini ditampilkan secara berurutan sesuai dengan jenis website yang ditemukan. Format laporan dapat dillihat pada **Tabel 5.1**.

Tabel 5.1 Format laporan

No	WordPress	PHP manual/framework
1	File Robots	File Robots
2	Directory Listing	Email
3	Celah pada versi dan plugin	SQL Injection
4	Username default	Local File Include
5	Full path disclosure	Cross Site Scripting

Laporan yang ditampilkan dalam sistem yang dikembangkan ini tidak selalu menampilkan hasil yang sama dengan format laporan pada tabel 5.1, hal ini tergantung dari celah keamanan yang ditemukan.

Proses pembuatan website dalam penelitian ini menggunakan framework Code Igniter. Proses eksekusi juga dilakukan melalui *framework* ini, berbeda dengan pembuatan laporan yang memanfaatkan *library* MPDF. Dikarenakan laporan hasil *scanning* ditampilkan dalam bentuk HTML, adanya plugin ini berfungsi untuk membuat file PDF melalui HTML sehingga bisa lebh memudahkan dalam menyimpan laporan.

5.4 Implementasi Sistem

5.4.1 Implementasi Deteksi File Robots

Seperti yang telah dirancang sebelumnya pendeteksian file robots dilakukan dengan mendeteksi file pada path /robots.txt kemudian mendeteksi status code, jika status code 200 maka file robots ditemukan, berikut adalah potongan kode untuk mendeteksi file robots.

```
def robots():

    try:
        req=urllib2.Request(url+"/robots.txt")
        response=urllib2.urlopen(req)
        the_page=response.read()
        print "<h2 style='font-family: Verdana, Helvetica, sans-
serif;font-weight:bold;font-size:   11pt;color:   #000000;width:
100%;background-color:#CCFFCC;text-align:          left;'>Found
Robots.txt</h2>",the_page
    except Exception, detail:
        print ''
```

Gambar 5.11 Source Code Deteksi File Robots

Dengan menggunakan *library* urllib2, sistem mencoba melakukan permintaan ke server dan membaca status code dari server. Jika ditemukan tampilkan seluruh isi dari *file robots* sebaliknya jika tidak ditemukan maka hanya muncul pesan kosong.

5.4.2 Implementasi Deteksi Jenis Website

Untuk proses pendeteksian jenis website, penulis menggunakan library request yang akan melakukan pengecekan terdahap di file yaitu:

- /wp-content
- /wp-login.php

Hasil pengecekan dari kedua file ini akan dimasukkan ke dalam bentuk parsing data dengan menggunakan library beautifulSoup.Jika sistem mendeteksi website jenis WordPress, maka hasil dari beautifulSoup akan diparsing untuk mengetahui versi Wordpress yang digunakan.

```
try:
        folderCheck=requests.get("%s/wp-content" % (url))
        folderCheck2=requests.get("%s/wp-login.php" % (url))
        if folderCheck or folderCheck2:
            html_page=urllib2.urlopen(url)
            soup=BeautifulSoup(html_page)
            try:
                #cek meta content apakah berisi string WordPress

hasil=str(soup.first('meta',attrs={'content':re.compile("WordPre
ss")})['content'])
            except:
```

```
                print "Tidak dapat mendeteksi versi WordPress"
                wpscan()
            else:
                #hasilTipe=hasil.split(' ')
                print "<b>Tipe Website :",hasil,"</b>"
                wpscan()
        else:
            print "Tipe Website : PHP/Framework"
            uniscan()
    except:
        print "Tidak dapat terhubung ke target. Silahkan periksa
Koneksi anda"
```

Gambar 5.12 Source Code Deteksi Jenis Website

Pada baris terakhir terdapat fungsi ping testing yang melakukan pengecekan apakah website sedang aktif atau tidak.

5.4.3 Implementasi Mekanisme Scanning WordPress

Dalam implementasi mekanisme scanning website jenis WordPress seperti dijeaskan pada perancangan bahwa dalam prosesnya akan menggunakan tool WPScan sehingga penulis akan memanfaatkan tool tersebut melalui pemrograman Python. Perintah tool WPScan akan dijalankan via *subprocess* dan kemudian akan diambil output dari subprocess. Output inilah yang digunakan sebagai data parsing.

```
def wpscan():
    command="wpscan -u %s --enumerate u" % (url)
    p = subprocess.Popen([command], stdout=subprocess.PIPE,
shell=True)

    (output, err) = p.communicate()
    vuln=re.findall('Title.*$',output,re.M)
    user=str(re.findall('Default
first.*$',output,re.M)).replace("'",'').replace('[','').replace(
']','').replace('"','')
    fpd=str(re.findall('Full Path
Disclosure.*$',output,re.M)).replace("'",'').replace('[','').rep
lace(']','').replace('"','')
    dir=str(re.findall('Directory listing is
enabled.*$',output,re.M)).replace('[','').replace(']','').replac
e("'",'').replace('Directory listing is enabled:
','').split(',')
    dir2= [x for x in dir if x]
    toStr = str(vuln)

akhir=toStr.replace("'",'').replace('[','').replace(']','').repl
ace('Title:
','').replace('\00\00\00','').replace('\00\00','').replace('\00'
,'')
    sp=akhir.split(',')
    if len(dir2) > 0:
        print "<h2 style='font-family: Verdana, Helvetica, sans-
serif;font-weight:bold;font-size: 11pt;color: #000000;width:
100%;background-color:#CCFFCC;text-align: left;'>Directory
Listing</h2>"
        for dr in dir2:
```

```
            url3=dr.lstrip().rstrip()
            print url3
    else:
        print ''

    print "<h2 style='font-family: Verdana, Helvetica, sans-
serif;font-weight:bold;font-size: 11pt;color: #000000;width:
100%;background-color:#CCFFCC;text-align:
left;'>Found",len(sp),"Vulnerabilites</h2>"
    for gg in sp:
        url2=gg.lstrip().rstrip()
        print "<a href='https://wpvulndb.com/search?text=%s'
target=_blank>%s</a>" % (str(urllib2.quote(url2)),str(url2))
    print """<h2 style='font-family: Verdana, Helvetica, sans-
serif;font-weight:bold;font-size: 10pt;color: #006400;width:
100%;background-color:#F0FFFF;text-align: left;'>Info and
Solution : <a href='https://wpvulndb.com'
target=_blank>https://wpvulndb.com</a></h2><h2 style='font-
family: Verdana, Helvetica, sans-serif;font-weight:bold;font-
size: 11pt;color: #000000;width: 100%;background-
color:#CCFFCC;text-align: left;'>Additional Information</h2>"""
    if len(user) > 0:
        print "&#9728; Username Default : admin</br><b><font
color=006400>Solution : Change Username</font></b></br>"
    else:
        print "&#9728; Username default : Tidak ditemukan"
    if len(fpd) > 0:
        print "&#9728; ",fpd
        print "<b><font color=006400>Solution : Add this code on
.htaccess : <br>&#9733; php_flag display_errors off <br>&#9733;
Options All -Indexes </font></b>"
    else:
        print "&#9728; Full Path Disclosure : Tidak ditemukan"
```

Gambar 5.13 Source Code Mekanisme Scanning WordPress

Berdasarkan kode diatas, dapat dilihat teknik parsing secara urut. Berikut adalah urutan parsing pada mekanisme ini:

1) Directory Listing
2) Celah Keamanan pada versi dan plugin Wordpress beserta link menuju WPVULNDB
3) Full Path Disclosure (FPD)
4) Username default 'admin'

5.4.4 Implementasi Mekanisme Scanning PHP Manual / framework

Dalam implementasi mekanisme Scanning PHP manual / framework tidak jauh berbeda dengan scanning pada website WordPress, perbedaan hanya terletak pada penggunaan tool yaitu menggunakan Uniscan.

```
def uniscan():
    command="sudo -S /usr/bin/uniscan -u %s -d" % (url)
    p =
subprocess.Popen([command],stdout=subprocess.PIPE,shell=True)
    (output, err) = p.communicate()
    #hasil = re.findall('[+].*$',output,re.M)
```

```
    sqli = str(re.findall('Blind SQL-
i].*$',output,re.M)).replace('[','').replace(']','').replace("'"
,'').replace('Blind SQL-i','').replace(': ','').split(',')
    lfi =
str(re.findall('LFI].*$',output,re.M)).replace('[','').replace('
]','').replace("'",'').replace('LFI','').split(',')
    email= str(re.findall('E-mail
Found:.*$',output,re.M)).replace('[','').replace(']','').replace
("'",'').replace('E-mail Found: ','').split(',')

    print "<h2 style='font-family: Verdana, Helvetica, sans-
serif;font-weight:bold;font-size: 11pt;color: #000000;width:
100%;background-color:#CCFFCC;text-align:
left;'>Found",len(email),"Email Address</h2>"
    for eml in email:
        print eml.lstrip()
    print "<h2 style='font-family: Verdana, Helvetica, sans-
serif;font-weight:bold;font-size: 11pt;color: #000000;width:
100%;background-color:#CCFFCC;text-align: left;'>Found SQL
Injection Vulnerability</h2>"
    for sql in sqli:
        urlsql=sql.lstrip().rstrip()
        print "<a href='%s' target=_blank>%s</a>" %
(str(urllib2.quote(urlsql)),str(urlsql))
    print "<h2 style='font-family: Verdana, Helvetica, sans-
serif;font-weight:bold;font-size: 11pt;color: #000000;width:
100%;background-color:#CCFFCC;text-align: left;'>Found Local
File Include Vulnerability</h2>"
    for lf in lfi:
        urllfi=lf.lstrip().rstrip()
        print "<a href='%s'target=_blank>%s</a>" %
(str(urllib2.quote(urllfi)),str(urllfi))
```

Gambar 5.14 Source Code Mekanisme Scanning PHP/Framework

5.4.5 Implementasi Website

Hal pertama yang diimplementasikan pada website adalah fungsi untuk menyimpan infomasi nama dan lokasi *file Python* (UBVSCAN). Fungsi dari informasi ini adalah agar memudahkan website dalam melakukan eksekusi file. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.

Nama Engine	Path
UBVSCAN	/var/www/html/ubvscan/ubvscan.py

Gambar 5.15 Menentukan Lokasi File

Pada **Gambar 5.15** terlihat nama *engine* yang digunakan adalah UBVSCAN dan lokasi file ini berada pada */var/www/html/ubvscan* dengan nama file *ubvscan.py*. Nama dan lokasi file dapat diubah sesuai dengan keinginan pengguna karena terdapat fitur edit. *File* UBVSCAN yang dikembangkan oleh penulis membutuhkan parameter agar dapat dijalankan sehingga dalam hal ini juga diimplementasikan fitur untuk menambah parameter seperti terlihat pada **Gambar 5.16**.

Nama Tool	Parameter
UBVSCAN	-u

Gambar 5.16 Menentukan Parameter

Parameter yang digunakan untuk menjalankan UBVSCAN adalah parameter "-u" dimana berfungsi sebagai perintah untuk memasukkan URL.

Setelah melakukan implementasi untuk informasi *file* dan parameter. Selanjutnya dilakukan implementasi untuk melakukan eksekusi *script*. Dalam eksekusi script diperlukan tiga parameter, yaitu lokasi file, parameter yang digunakan, dan URL website yang akan dijadikan sebagai target *scanning.* Berikut adalah hasil implementasi untuk ketiga parameter tersebut.

```
            $host = (string) $this->input->post("host", true);
            $p0 = $this->tool_model->get_by();
            foreach($p0 as $p1) {
                $param = $this->parameter model->get by(["tool"
=> $p1["id"]]);
                foreach($param as $p) {
                    $cmd[] = $p1["path"] . " " . $p["param"] . "
" . $host;
                }
```

Gambar 5.17 Source Code Tiga Parameter Eksekusi

Pada baris ke-1 merupakan variabel yang menyimpan input URL dari user,baris ke-2 berfungsi untuk mengambil data informasi lokasi *script* dan baris ke-4 berfungsi untuk mengambil informasi parameter yang digunakan. Ketiga parameter ini kemudian disimpan pada variabel *$cmd* dengan urutan lokasi *file*, paramater dan URL. Parameter ini kemudian digunakan dalam proses eksekusi pada kode di bawah ini.

```
            foreach($cmd as $c) {
                $ret = shell_exec($c . " 2>&1");

                $out .= "<pre>" . $ret . "</pre>";
            }

            foreach($p0 as $p1) {
                $this->laporan_model->make([
                    "tool" => $p1["nama"],
                    "host" => $host,
                    "value" => $out
                ]);
```

Gambar 5.18 Source Code Ekseksusi File Python

Pada kode diatas, digunakan fungsi *shell_exec* dari PHP untuk melakukan eksekusi pada tiga parameter yang telah dijelaskan sebelumnya dan hasilnya disimpan dengan format *tool* yang digunakan, URL dan hasil *scanning* pada *database* dan hasil ini juga langsung ditampilkan dalam format laporan yang telah dirancang seperti terlihat pada gambar di bawah ini.

Gambar 5.19 Laporan *Scanning*

Format laporan pada **Gambar 5.19** disimpan dalam format HTML. Berdasarkan perancangan, diperlukan fungsi agar laporan dapat disimpan dalam format PDF. Disini penulis menggunakan *library MPDF*. Untuk lebih jelasnya dapat dilihat pada kode di bawah ini.

```
class Mypdf {
    public function generate($data=[]) {
        $mpdf = new mPDF("utf-8", "A4", 0, "", 12.7, 12.7, 14,
12.7, 8, 8);
        $mpdf->SetDisplayMode("fullpage");
        $mpdf->list_indent_first_level = 0;

        $mpdf->WriteHTML($data["value"]);
        return $mpdf->Output();
    }
```

Gambar 5.20 Source Code Simpan Laporan Berbentuk PDF

Pada *class MyPdf* terdapat fungsi untuk membuat file PDF dengan mengambil data dari hasil scanning yang tersimpan pada database,data dengan format HTML akan dikonversi menjadi file PDF dengan format yang digunakan dalam membuat PDF adalah *encoding utf-8*, ukuran kertas A4 *fullpage* dan ukuran *margin* 0.

BAB 6 PENGUJIAN DAN ANALISIS

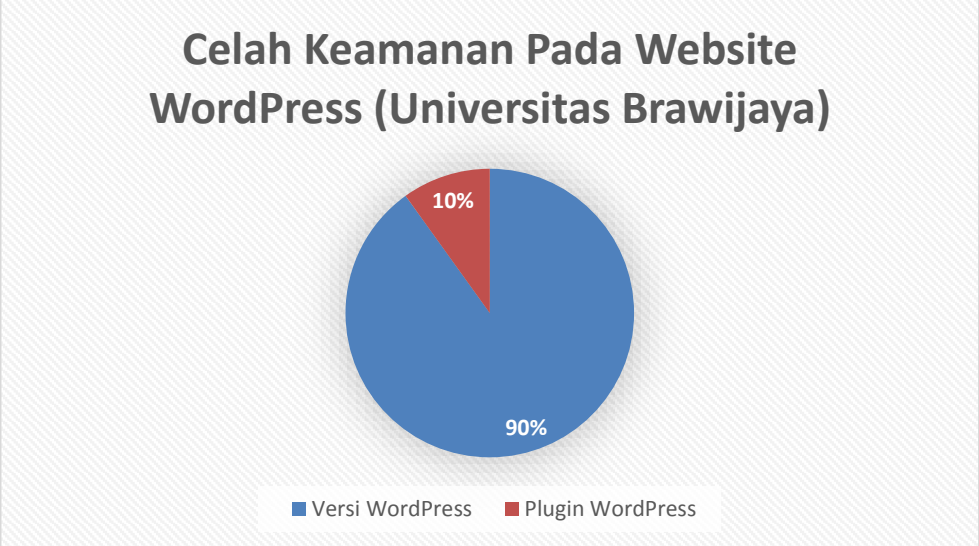
Berdasarkan prosedur pengujian pada Bab 3, dilakukan pengujian dengan melakukan *scanning* pada semua data uji berupa website-website di Universitas Brawijaya yang telah direplikasi.

Terdapat 2 metode pengujian yang dilakukan dalam penetian ini, yaitu pengujian deteksi celah keamanan yang dilakukan pada website yang menggunakan WordPress dan website PHP manual/framework dan metode pengujian selanjutnya adalah pengujian akurasi sistem.Pengujian akurasi sistem dilakukan dengan membandingkan hasil deteksi celah keamanan dari sistem yang telah dibangun dengan *tool* deteksi celah keamanan sejenis yang memiliki tujuan yang sama yaitu untuk mendeteksi celah keamanan pada website.

6.1 Hasil dan Analisis Deteksi Celah Keamanan

Terdapat 2 skenario yang dilakukan dalam pengujian ini, yaitu pengujian pada website yang menggunakan WordPress dan pengujian pada website yang menggunakan PHP manual/framework.Hasil pengujian lebih difokuskan pada website yang menggunakan WordPress dikarenakan data replikasi yang sesuai dengan data website asli.

6.1.1 Hasil pengujian dan analisis website WordPress



Gambar 6.1 Diagram Celah Keamanan Pada Website WordPress

Pada skenario pertama, dilakukan pengujian pada website yang menggunakan WordPress.Dari 54 data website replikasi WordPress yang di uji, 53 website ditemukan memiliki celah keamanan yang terdapat pada versi WordPress dan pada versi plugin yang digunakan.Dari hasil pengujian ini, dilakukan analisa untuk mengetahui celah keamanan yang dominan pada website-website di Universitas Brawijaya khususnya website yang menggunakan WordPress, hasil analisis dapat dilihat pada **Gambar 6.1**.

Berdasarkan diagram pada **Gambar 6.1**, didapatkan persentase 90% pada versi WordPress yang digunakan sedangkan celah pada *plugin* didapatkan persentase 10%.Dari hasil persentase tersebut, didapatkan kesimpulan bahwa website-website yang berada di Universitas Brawijaya khususnya yang menggunakan WordPress memiliki celah keamanan yang dominan terdapat pada versi yang digunakan.

Celah keamanan pada WordPress sering terjadi karena website yang jarang diperbaharui (*update*) khususnya pada versi yang digunakan. WordPress akan menampilkan notifikasi jika terdapat *update* terbaru, namun masih banyak yang belum menyadari pentingnya memperbaharui WordPress. Sama seperti plugin pada WordPress yang juga membutuhkan *update* jika terdapat versi baru dari plugin tersebut,WordPress akan memberikan notifikasi untuk segera melakukan *update*, plugin pada WordPress sering menjadi target bagi para *attacker* dalam melakukan aksi hacking hal ini terjadi karena plugin WordPress dapat dibuat oleh pihak ketiga.

6.1.2 Hasil pengujian dan analisis website PHP manual/framework

Pada skenario kedua, dilakukan pengujian pada website yang menggunakan PHP manual /framework. Dilakukan scanning pada semua data uji. Dari 6 website data uji, 3 website ditemukan memiliki celah keamanan. Berikut adalah hasil pengujian yang telah dilakukan.

Tabel 6.1 Hasil Pengujian Data uji Ke-1

URL	http://localhost/mtic	
Jenis Web	PHP manual / framework	
Celah	**Deskripsi**	**Solusi**
SQL Injection : TourInformation.php?id=	Celah SQL injection yang terjadi pada file TourInformation.php dimana parameter id tidak di-filter sehingga bisa diinjeki dengan perintah SQL	Gunakan sanitasi karakter spesial pada parameter id

Pada Website data uji pertama seperti terlihat pada tabel 6.1, sistem menemukan celah SQL Injection yang terjadi pada file *Tourinformation.php* tepatnya pada parameter **id**, artinya kita bisa melakukan SQL Injection pada URL di gambar berikut ini :

http://localhost/mtic/TourInformation.php?id=[id]

Parameter id pada URL diatas tidak dilakukan filterisasi sehingga penyerang dapat menginputkan kode SQL Injection, berdasarkan celah yang ditemukan,solusi yang dapat digunakan untuk mengatasi celah ini adalah dengan melakukan filtering atau melakukan sanitasi input pada parameter id.

Tabel 6.2 Hasil Pengujian Data Ke-3

URL	http://localhost/sete	
Jenis Web	PHP manual / framework	
Celah	**Deskripsi**	**Solusi**
SQL Injection : cara.php?id=	Celah SQL injection yang terjadi pada file cara.php dimana parameter id tidak di-filter sehingga bisa diinjeki dengan perintah SQL	Gunakan sanitasi karakter spesial pada parameter id
Local File Include : about.php?id=	Celah Local File Inclusion yang terjadi pada about.php dimana parameter id tidak di-filter sehingga attacker dapat melakukan *code execution*	Gunakan sanitasi karakter spesial pada parameter id dan lakukan validasi data pada parameter

Pada website data ketiga seperti terlihat pada **Tabel 6.2**, sistem menemukan dua celah keamanan, yaitu *SQL Injection* dan *Local File Include*.Celah SQL injection terdapat pada file *cara.php* tepatnya pada parameter id yang bisa diakses melalui URL :

http://localhost/sete/cara.php?id=[id]

Selanjutnya celah *Local File Include* yang terdapat pada file *about.php* tepatnya pada parameter **id** yang bisa diakses melalui URL:

http://localhost/sete/about.php?id=[id]

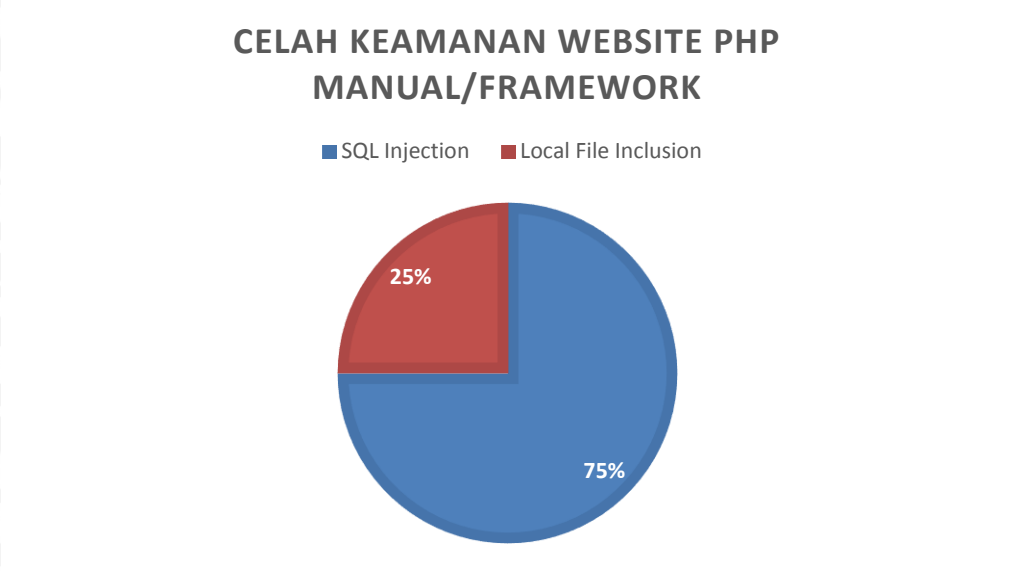
Kedua celah ini tidak melakukan filtering pada parameter id sehingga solusinya adalah melakukan *filtering* atau sanitasi terhadap semua input pada parameter id.

Tabel 6.3 Hasil Pengujian Data Ke-5

URL	http://localhost/mtic	
Jenis Web	PHP manual / framework	
Celah	**Deskripsi**	**Solusi**
SQL Injection : index.php?id=	Celah SQL injection yang terjadi pada file index.php dimana parameter id tidak di-filter sehingga bisa diinjeki dengan perintah SQL	Gunakan sanitasi karakter spesial pada parameter id

Pada website data kelima seperti terlihat pada tabel 6.3, sistem menemukan celah *SQL injection* pada file *index.php* tepatnya pada parameter **id**. Sama seperti celah pada data uji lainnya, pada parameter **id** tidak ada filterisasi sehingga solusinya adalah melakukan filterisasi atau sanitasi input pada parameter **id**.

Hasil pengujian kemudian dianalisis untuk mengetahui celah apa yang dominan dan hasil dapat dilihat pada **Gambar 6.2**.



Gambar 6.2 Celah Keamanan Dominan Website PHP manual

Berdasarkan hasil analisis, didapatkan persentase 75% pada celah *SQL Injection* dan 25% pada *Local File Include*. Hal ini menunjukkan hasil pengujian pada website yang menggunakan PHP manual/framework lebih dominan memiliki celah *SQL Injection*.Tidak seperti website WordPress yang memiliki data replikasi yang lengkap, pengujian ini tidak dapat dijadikan sebagai acuan dalam menentukan celah keamanan di Universitas Brawijaya namun hanya digunakan untuk mengetahui apakah sistem sudah dapat melakukan deteksi celah keamanan sesuai dengan perancangan yang telah direncanakan.

Hasil pengujian deteksi celah keamanan pada dua jenis website menunjukkan bahwa sistem dapat berjalan dengan baik dilihat dari celah keamanan yang ditemukan.Hasil *scanning* lebih dominan terlihat pada website jenis WordPress daripada website yang menggunakan PHP manual / framework hal ini disebabkan banyaknya celah keamanan yang ditemukan pada website yang menggunakan WordPress.

6.2 Hasil dan Analisis Akurasi Sistem

Sama halnya dengan pengujian deteksi celah keamanan dalam pengujian akurasi sistem terdiri dari 2 skenario percobaan yaitu pengujian pada website yang menggunakan CMS wordPress dan website yang menggunakan PHP manual/framework.

Website dalam pengujian dipilih secara acak dari data replikasi website,digunakan 2 website yang menggunakan WordPress dan 2 website yang menggunakan PHP manual / framework sehingga total data uji berjumlah 4 website.Berikut adalah data website uji yang telah dipilih secara acak.

Tabel 6.4 Data Pengujian

No	Domain	Keterangan	Jenis Web
1	http://localhost/fapet	Fakultas Peternakan	WordPress
2	http://localhost/teknik	Fakultas Teknik	WordPress
3	http://localhost/sete	-	PHP manual / framework
4	http://localhost/injector	-	PHP manual / framework

Proses pengujian pada data uji diatas memerlukapan parameter berupa celah yang akan dideteksi oleh masing-masing *tools* dikarenakan banyaknya celah keamanan pada website sehingga dalam hal ini penulis membatasinya. Untuk website yang menggunakan WordPress, maka pendeteksian celah keamanan dilakukan pada parameter berikut ini:

- File Robots
- Deteksi celah WordPress

Deteksi celah WordPress pada parameter diatas, mencakup semua celah keamanan baik pada versi yang digunakan dan plugin yang digunakan. Sedangkan untuk website yang menggunakan PHP manual/*framework*, pendeteksian dilakukan pada parameter berikut ini:

- File Robotsj
- SQL Injection
- Cross Site Scriting (XSS)
- Local File Include

6.2.1 Hasil pengujian dan analisis website WordPress

Pada skenario pertama dilakukan pengujian pada website yang menggunakan WordPress. Berikut ini merupakan hasil pengujian akurasi yang dilakukan pada website yang menggunakan WordPress.

Tabel 6.5 Hasil perbaningan data uji pertama

Website	http://localhost/fapet			
Celah	**UBVSCAN**	**Nessus**	**Nikto**	**Acunetix**
File Robots	✓	✓	✓	✓
Celah WordPress	✓			✓

Pada data uji pertama seperti terlihat pada **Tabel 6.5**, ditemukan dua jenis celah keamanan, yaitu file robots dan celah pada website WordPress. Semua tools berhasil mendeteksi file robots namun hanya dua tools yang dapat mendeteksi celah keamanan pada WordPress yaitu UBVSCAN dan Acunetix. Kedua *tools* ini memiliki hasil yang berbeda dalam mendeteksi celah keamanan dari segi jumlah. Berdasarkan penemuan celah keamanan pada versi dan plugin yang digunakan, pada UBVSCAN ditemukan 22 celah keamanan,sedangkan Acunetix menemukan 10 celah keamanan.

Tabel 6.6 Hasil perbandingan data uji kedua

Website	http://localhost/teknik			
Celah	**UBVSCAN**	**Nessus**	**Nikto**	**Acunetix**
File Robots	✓	✓	✓	✓
Celah WordPress	✓			✓

Seperti data uji pertama, pada data uji kedua, perbedaan pada kedua tools dari UBVSCAN dan Acunetix adalah jumlah celah keamanan yang ditemukan. Dalam pengujian data uji kedua, pada UBVSCAN ditemukan 14 celah keamanan sedangkan pada Acunetix ditemukan 7 celah keamanan.

Berdasarkan kedua tabel diatas, ditemukan perbedaan jumlah celah keamanan pada tools UBVSCAN dan Acunetix, dari hasil perbandingan jumlah celah keamanan, UBVSCAN lebih banyak menemukan celah keamanan daripada Acunetix. Peneliti melakukan analisa lebih lanjut dan menemukan bahwa celah keamanan yang ditemukan oleh Acunetix bersumber dari WPVULNDB namun data yang digunakan tidak *update* berbeda dengan UBVSCAN dimana data yang digunakan dapat di-*update* melalui tool WPScan sehingga celah keamanan yang ditemukan dapat lebih banyak.

Pada hasil pengujian ini didapatkan kesimpulan bahwa UBVSCAN lebih unggul dalam mendeteksi celah keamanan pada website yang menggunakan WordPress.

6.2.2 Hasil pengujian dan analisis website PHP Manual/Framework

Pada skenario kedua, dilakukan pengujian pada website yang menggunakan PHP manual/framework. Berikut ini merupakan hasil pengujian akurasi yang dilakukan pada website yang menggunakan PHP manual/framework.

Tabel 6.7 Hasil perbandingan data uji ketiga

Website	http://localhost/sete			
Celah	**UBVSCAN**	**Nessus**	**Nikto**	**Acunetix**
File Robots	✓	✓	✓	✓
SQL Injection	✓	✓		✓

Cross Site Scripting (XSS)		✓		✓
Local File Include				

Pada data uji ketiga seperti terlihat pada **Tabel 6.7**, ditemukan 3 celah keamanan, yaitu file robots, SQL injection dan Cross Site Scripting (XSS). Dari hasil perbandingan, semua tools dapat mendeteksi file robots, kemudian celah SQL Injection yang dapat terdeteksi oleh ketiga tool yaitu UBVSCAN, Nessus dan Acunetix sedangkan Nikto tidak mendapatkan hasil apapun mengenai SQL injection. Dalam mendeteksi celah cross site scripting (XSS), hanya Acunetix dan Nessus yang dapat menemukan celah tersebut sedangkan tools lainnya tidak menghasilkan apapun terkait celah Cross Site Scripting (XSS). Tidak seperti hasil pengujian pada website WordPress, pada data keempat tidak memiliki hasil yang sama dengan data ketiga, hasil pengujian pada data keempat dapat pada **Tabel 6.8**.

Tabel 6.8 Hasil perbandingan data uji keempat

Website	http://localhost/injector			
Celah	**UBVSCAN**	**Nessus**	**Nikto**	**Acunetix**
File Robots				
SQL Injection	✓	✓		✓
Cross Site Scripting (XSS)		✓		✓
Local File Include				

Pada data uji keempat seperti terlihat pada **Tabel 6.8**, didapatkan 2 celah keamanan, yaitu SQL Injection dan Cross Site Scripting (XSS).Dari hasil perbandingan, celah SQL injection ditemukan oleh UBVSCAN,Nessus dan Acunetix, selanjutnya celah Cross Site Scripting yang hanya ditemukan oleh tool Acunetix dan Nessus sedangkan *tools* lainnya tidak mendapatkan hasil apapun terkait Cross Site Scripting (XSS).

Pada **Tabel 6.5** dan **Tabel 6.6** didapatkan hasil yang sama dimana semua *tools* dapat mendeteksi file *robots* pada website, terdapat dua *tools* yang dapat mendeteksi celah keamanan pada WordPress yaitu tool UBVSCAN yang dikembangkan oleh penulis dan Acunetix namun UBVSCAN lebih unggul dalam menemukan celah keamanan, hal ini membuktikan bahwa pendeteksian celah keamanan pada jenis website cukup efektif khususnya pada website jenis CMS. Website jenis CMS menggunakan desain yang sama namun terdapat versi yang

beragam,versi yang beragam ini merupakan *update* dari penyedia CMS untuk melakukan perbaikan bug serta peningkatanan fitur dari versi sebelumnya, berdasarkan versi-versi ini dapat dibuat database untuk melakukan scanning pada jenis CMS tertentu contohnya dalam pengujian ini menggunakan database dari WPVULNDB untuk melakukan scanning pada website yang menggunakan WordPress. Namun pendeteksian celah keamanan berdasarkan database ini memliki kelemahan dimana informasi pada database yang terbatas sehingga jika terdapat celah keamanan yang terbaru maka sistem tidak dapat melakukan deteksi celah keamanan.Berbeda dengan *tool* Nessus dan Acunetix yang menganggap semua website sama sehingga akan melakukan scanning secara menyeluruh dan tidak hanya bergantung pada database yang memungkinkan dapat mendeteksi celah keamanan baru pada sebuah website.

Pada **Tabel 6.7** dan **Tabel 6.8**, *tool* Acunetix terlihat lebih dominan dalam mendeteksi celah kemanan, dari semua *tools* yang digunakan, hanya Acunetix yang dapat mendeteksi celah XSS pada website data uji ketiga. *Tool* UBVSCAN tidak dapat mendeteksi celah XSS sehingga dibutuhkan penambahan *tool* khusus untuk mendeteksi celah XSS seperti.

BAB 7 KESIMPULAN

7.1 Kesimpulan

Berdasarkan hasil penelitian ini, dapat disimpulkan beberapa hal mengenai sistem deteksi celah keamanan yang telah di uji pada beberapa website.

1. Sistem Deteksi dapat berjalan sesuai dengan perancangan yang dilakukan. Dari 54 website data replikasi, sistem yang dibangun menemukan celah keamanan pada 53 website.Celah keamanan pada WordPress banyak ditemukan pada versi WordPress dan versi plugin yang digunakan.Sedangkan pengujian pada PHP manual/framework dari 6 data website, hanya 3 website yang ditemukan memiliki celah keamanan.
2. Deteksi celah keamanan berdasarkan jenis website memiliki kelebihan terutama pada website WordPress dikarenakan celah keamanan pada WordPress yang beragam sesuai dengan versi yang digunakan bersumber dari WPVULNDB yang menyimpan *database* celah keamanan pada WordPress.Namun metode ini memiiliki kelemahan dimana terbatasnya informasi berupa *database* yang digunakan untuk mendeteksi celah keamanan.
3. Berdasarkan hasil perbandingan dengan *tools* sejenis, sistem yang dibangun memiliki kelebihan dalam mendeteksi celah keamanan pada WordPress namun memiliki kelemahan dalam mendeteksi celah keamanan pada website PHP manual/framework terutama pendeteksian celah Cross Site Scripting (XSS).

7.2 Saran

Berikut beberapa saran yang diharapkan dalam pengembangan sistem ini.

1. Sistem yang telah dibuat perlu pengembagnan lebih lanjut dalam mendeteksi berbagai jenis website, tidak terbatas pada CMS WordPress dan PHP Manual/framework.
2. Diperlukan pemanfaatkan *search engine* yang perlu diimplementasikan pada sistem untuk mencari celah keamanan melalui *search engine* dan juga diperlukan fitur *deep scanning* agar sistem dapat melakukan *scanning* secara optimal.

DAFTAR PUSTAKA

Akamai. 2014. *Akamai's State of The Internet Q1 2014 Report | Volume 7 Number 1*. Akamai.

Kennedy, D., O'Gorman, J., Kearns, D., & Aharoni, M. 2011. *Metasploit The Penetration Tester's Guide*. San Francisco: William Pollock.

UBHosting. 2014. *Aturan Webhosting*. [online] UBHosting. Tersedia di: <https://hosting.ub.ac.id/aturan-webhosting/> [Diakses 18 September 2015].

UBHosting. 2014. *Penjelasan Formulir Registrasi*.[online] UBHosting. Tersedia di: <https://hosting.ub.ac.id/penjelasan-formulir-registrasi/> [Diakses 18 September 2015].

Tarigan, I. A. 2015. *BIN, BAIS, BNN, Brimob dan KPK Pernah Diincar Hacking Team sebagai Klien*. [online] Metrotvnews.com. Tersedia di: <http://teknologi.metrotvnews.com/read/2015/07/13/146738/bin-bais-bnn-brimob-dan-kpk-pernah-diincar-hacking-team-sebagai-klien> [Diakses 26 September 2015].

Amalia, E. I. 2015. *Data Internal Hacking Team Kini Dapat Diakses via WikiLeaks*.[online] Metro TV News.Tersedia di: <http://teknologi.metrotvnews.com/read/2015/07/13/146732/data-internal-hacking-team-kini-dapat-diakses-via-wikileaks> [Diakses 26 September 2015].

Briggs, A. 2012. *Hello! Python*. Shelter Island: Manning Publications.

Kim, P. 2015. *The Hacker Playbook 2 Practical Guide To Penetration Testing*. North Charleston: Secure Planet LLC.

McQuade, K. 2014. *Open Source Web Vulnerability Scanners: The Cost Effective Choice?*. Arlington: Marymount University.

Bairwa, S., Mewara, B., & Gajrani, J. 2014. *Vulnerability Scanners: A Proactive Approach To Assess Web Application Security*. Ajmer : Government Engineering College.

Gupta, H., Sheetlani, J., & Gupta, P. 2015. *Comparative Study of Vulnerability Assessment Tools for Network Security*.

SecTools. (n.d.).*SecTools.Org: Top 125 Network Security Tools*.[online] SecTools.Tersedia di: <http://sectools.org> [Diakses 21 April 2016].

Kennedy, D., O'Gorman, J., & Aharoni, M. 2011. *Metasploit The Penetration Tester's Guide.* San Francisco: No Starch Press, Inc.

SANS Institute. 2011. Robots.txt.

OWASP. 2014. *Testing for Local File Inclusion*. [online] OWASP. Tersedia di: <https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion> [Diakses 25 April 2016]

OWASP. 2012. *Full Path Disclosure*. [online] OWASP. Tersedia di<https://www.owasp.org/index.php/Full_Path_Disclosure>[Diakses 25 April 2016)

Lekies, S., Stock, B., & Johns, M. 2014. *A tale of the weaknesses of current client-side XSS Filtering.*

Mitchell, A. 2015. *Using WPScan: Finding WordPress Vulnerabilities*. [online] Sucuri. Terdedia di <https://blog.sucuri.net/2015/12/using-wpscan-finding-wordpress-vulnerabilities.html>[Diakses 5 Juni 2016]

Ports. (2014). *Uniscan*. [online] Kali Tools. Tersedia di <http://tools.kali.org/web-applications/uniscan>[Diakses 5 Juni 2016]