# KINERJA *MODSECURITY TECHNICAL REPORT* (STUDI KASUS: PENCEGAHAN TERHADAP SERANGAN *SQL INJECTION*)

**Farid Ridho**
**Dosen Sekolah Tinggi Ilmu Statistik**

***Abstract***

*Several Measures are impelemented in web application security lifecycle such as Secure Development, Secure Deployment and Secure Operation. In secure operation section, a web application that has been through the stages of development and testing will soon enter production phase. At this stage it will be applied to Web Application Firewall (WAF) that meant to protect application from a malicious request.*

*The purpose of this research is to explore ModSecurity WAF implementation. WAF ModSecurity is a free, open source application that can be used to make the filter to requests which occur on a web application including a request containing SQL Injection commands. Another aim is to see whether the ModSecurity installation on a web server affect the performance of the web server.*

*From the test results concluded that ModSecurity can filter SQL injection and installation of ModSecurity does not significantly affect the performance of the web server.*

***Keywords**: Web Application Security, WAF, Modsecurity Performance, SQL Injection*

## I. PENDAHULUAN

Saat ini statistik serangan terhadap aplikasi web semakin meningkat setiap tahunnya (Sumantri, 2012). Peningkatan serangan itu terjadi baik dalam jumlah serangan maupun intensitas serangan. Jenis serangan yang sering terjadi masih didominasi serangan dengan teknik *SQL injection*, *Cross Side Scripting* (XSS) dan juga *Distributed Denial of Service* (DDOS) (OWASP, 2013).

Salah satu teknik untuk mencegah adanya serangan pada aplikasi web adalah dengan melakukan pembuatan aplikasi web yang mempertimbangkan aspek keamanan atau yang sering disebut sebagai secure programming. Pendekatan lainnya adalah dengan melakukan *secure deployment* dan *secure operation* (Mogul & Lane, 2009).

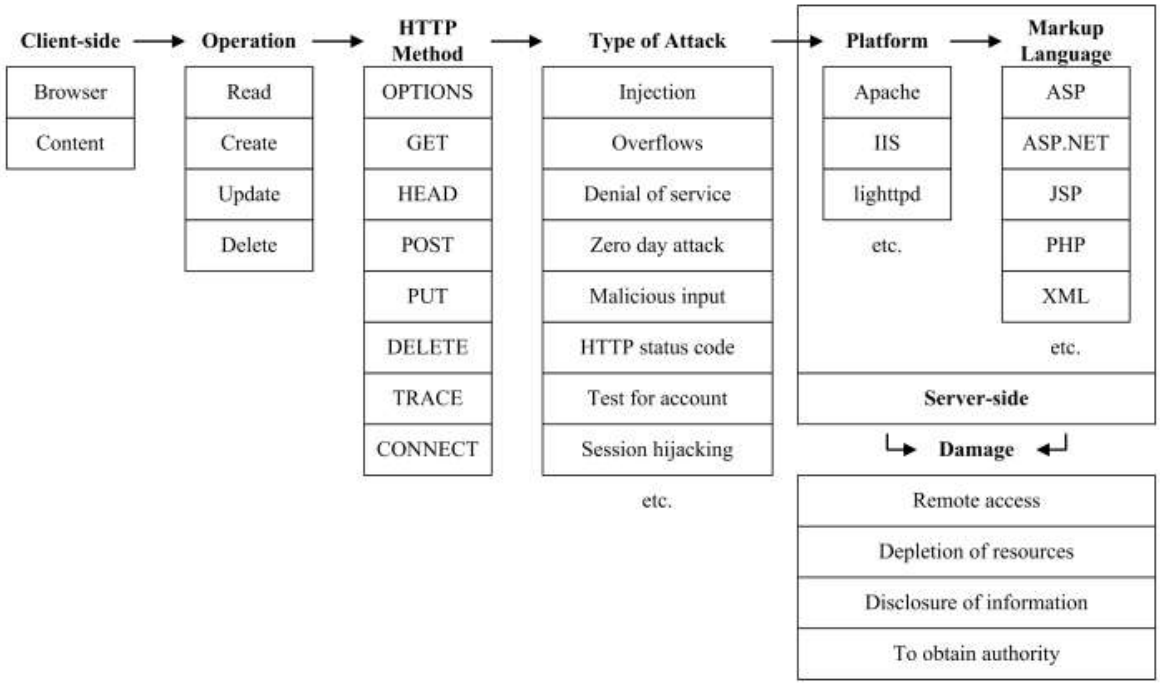
Paper ini akan membahas tentang bagaimana meningkatkan keamanan aplikasi web dengan cara menerapkan *secure operation*. Salah satu caranya yaitu dengan memasang *Web Aplication Firewall* (*WAF*) pada web server. Penerapan WAF akan mengurangi resiko keamanan yang terjadi pada aplikasi web akibat kesalahan pada tahap *secure development* dan *secure deployment*. Implementasi WAF ini juga diharapkan tidak terlalu berpengaruh terhadap performa dari *Web server* dalam melayani *request* dari pengguna. Karena itu akan dibahas juga sejauh mana sebuah WAF mampu diandalkan untuk mencegah serangan pada aplikasi web.

## II. METODOLOGI

### A. Kajian Pustaka

#### Taksonomi serangan pada aplikasi web

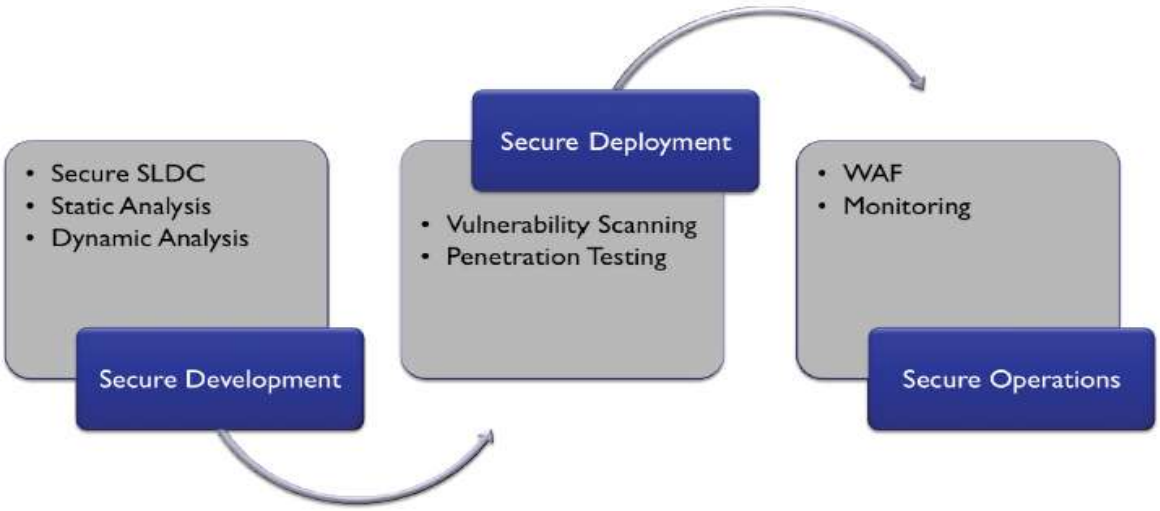
Perkembangan teknologi web saat ini semakin pesat. Kebutuhan untuk berbagi informasi melalui jejaring sosial, menjalankan bisnis perusahaan dan juga melakukan layanan lainnya, menjadikan website menjadi lebih rentan untuk diserang. Berbagai jenis serangan terjadi pada aplikasi web sehingga menimbulkan kerugian yang tidak sedikit. Jenis-jenis serangan ini dapat dibagi menjadi beberapa tipe serangan. Gambar 2.1 menjelaskan taksonomi dari serangan yang muncul terhadap aplikasi web. Tipe serangan seperti *injection*, *overflows*, *denial of service* (*DOS*) memang menjadi serangan yang paling sering muncul. Penjelasan lebih rinci mengenai taksonomi serangan pada aplikasi web dapat merujuk pada paper yang tertera di daftar pustaka (Álvarez & Petrovic, 2003).




**Gambar 1. Taksonomi serangan pada aplikasi Web (Álvarez & Petrovic, 2003)**

## Keamanan Aplikasi Web

Mogull dan Lane (2009) membuat daur hidup keamanan aplikasi web. Aplikasi web dibangun berdasarkan siklus daur hidup keamanan aplikasi yang dibagi menjadi 3 bagian besar seperti terlihat pada gambar 2.




**Gambar 2 Siklus Hidup Keamanan Aplikasi Web (Mogul & Lane, 2009)**

***Secure Development***

*Secure development* adalah bagaimana membangun aplikasi web dengan menerapkan prinsip keamanan. Beberapa pendekatan untuk melakukan *secure development* adalah dengan menerapkan *secure software development lifecycle* (*SDLC*), *static analysis* dan *dynamic analysis.*

***Secure Deployment***

Setelah semua tahap dalam pengembangan aplikasi selesai, tahap berikutnya adalah melakukan pengujian dan juga validasi. Tahap ini dilakukan untuk memastikan bahwa aplikasi tidak memiliki celah keamanan yang serius. Pengujian dan validasi dapat dilakukan dengan mengunakan metode *vulnerability assessment* dan *penetration testing*.

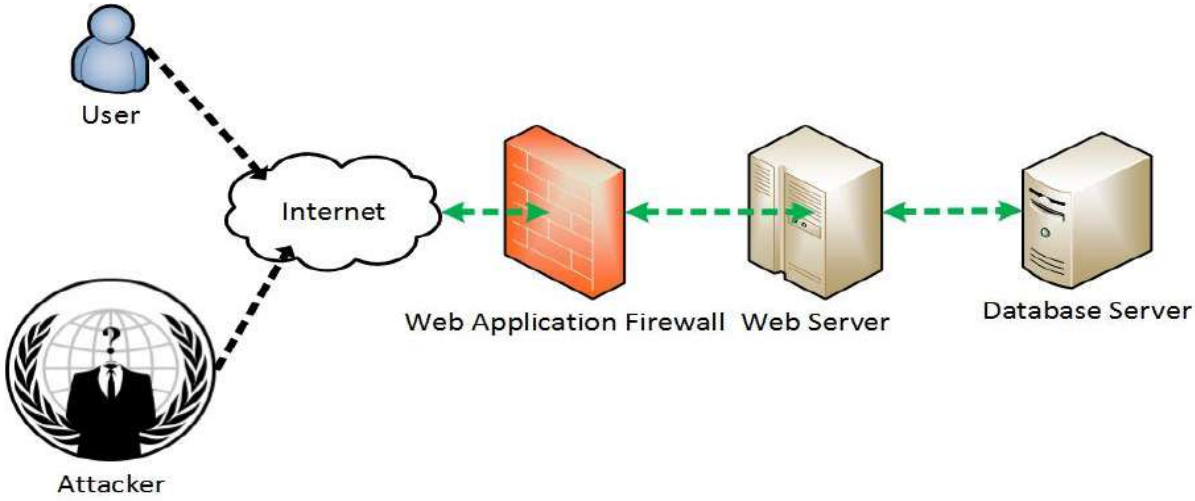
1. *Vulnerability Assessment*, melakukan *scanning* pada aplikasi web untuk mengetahui celah keamanan.
2. *Penetration Testing*, adalah proses untuk membobol aplikasi untuk menentukan celah keamanan dan resiko yang ditimbulkannya. Proses *vulnerability assessment* digunakan menemukan celah keamanan sedangkan *penetration testing* memerika semua lubang untuk mengukur dampak.

***Secure Operation***

*Secure operation* adalah bagaimana kita meningkatkan keamanan aplikasi web dengan cara mengimplementasikan perangkat keamanan seperti *web application firewall* (*WAF*) maupun aplikasi monitoring lainnya pada saat aplikasi web telah dioperasikan.

***Web Application Firewall****,*

Menurut *Web Application Security Consortium* (*WASC*) *web application firewall* (*WAF*) diartikan sebagai sebuah perangkat perantara, yang berada antara *web client* dan *web server*, menganalisis pesan pada OSI *Layer*-7 ketika terjadi pelanggaran dalam kebijakan keamanan yang telah ditentukan (*Web Security Glossary*, 2013). Sebuah *web application firewall* digunakan sebagai perangkat keamanan untuk melindungi server web dari serangan. Gambaran dari sebuah *web application firewall* dapat dilihat pada Gambar 3.

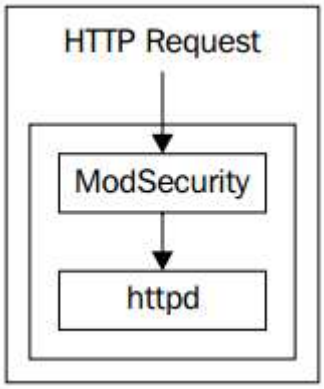



**Gambar 3 Gambar sebuah *web application firewall***

Banyak WAF yang tersedia di pasaran baik itu yang berbayar seperti *Cloudflare* dan *Incapsula* maupun yang gratis seperti *ModSecurity*. Pada penelitian kali ini akan digunakan WAF dan *open source* yaitu *ModSecurity*. Tim *Zero Science Lab* telah membandingkan kelebihan dan kekurangan ketiga jenis WAF ini (S. Petrushevski & Cabrera, 2013).

***ModSecurity***

Seperti umumnya *firewall*, *ModSecurity* melakukan filter terhadap data yang masuk dan data yang keluar pada sebuah *web server*. *ModSecurity* mampu menghentikan sebuah *traffic* yang dicurigai sebagai *malicious request*. *ModSecurity* juga memiliki banyak fitur lainnya seperti melakukan *logging* transaksi yang terjadi pada protokol HTTP dan *content injection* (Mische, 2009).



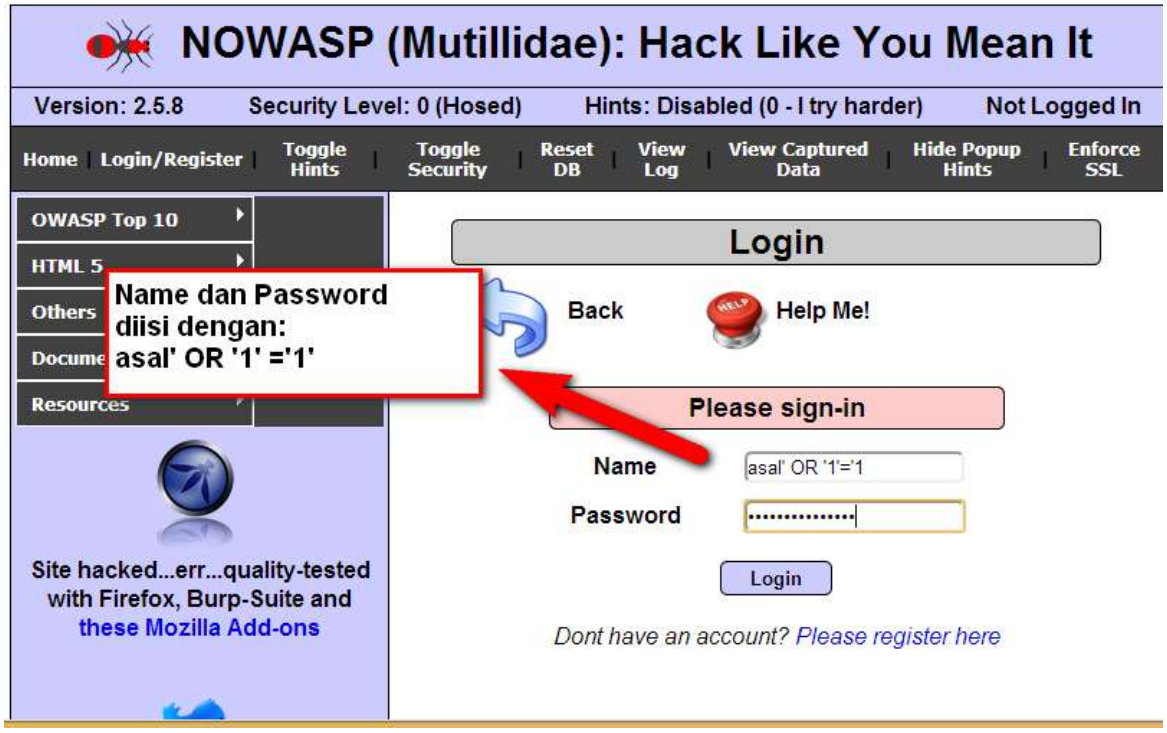

**Gambar 4 Arsitektur *ModSecurity* pada *web server***

Dari Gambar 4 terlihat bahwa *ModSecurity* berada di antara *web server* dan *HTTP request* yang dikirimkan oleh pengguna.

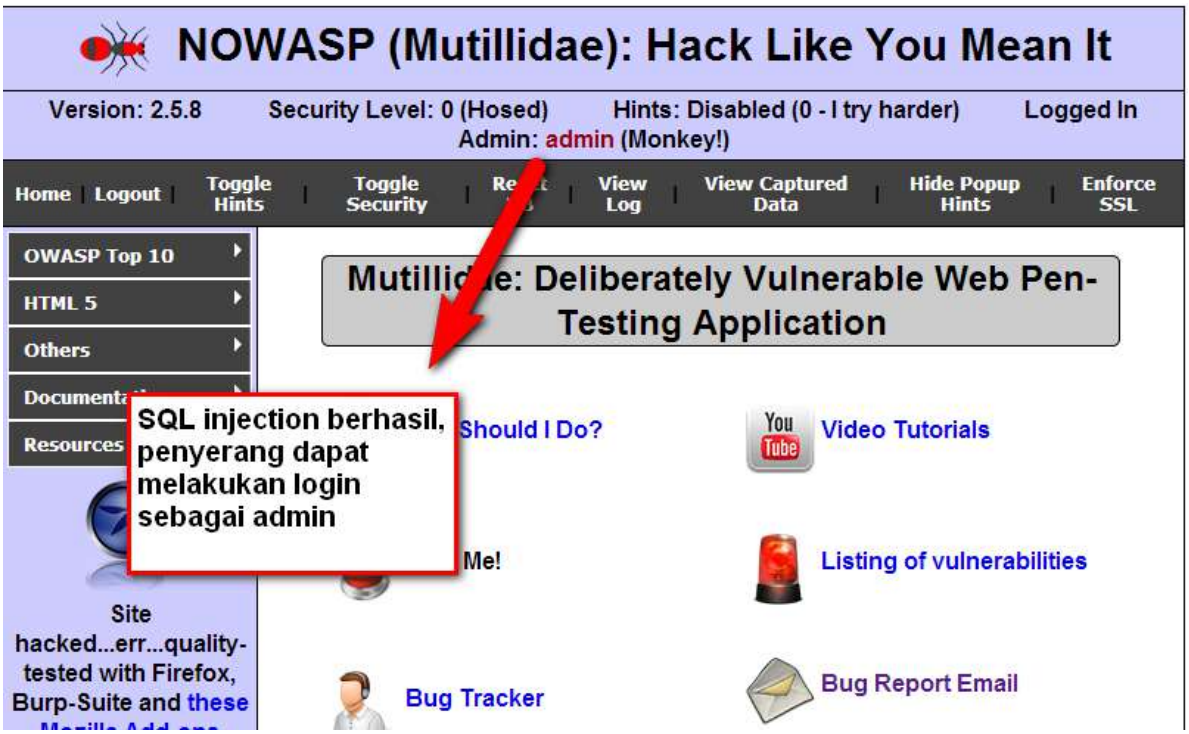
***Mutillidae***

Untuk pengujian bagaimana *ModSecurity* melakukan pencegahan serangan terhadap aplikasi web, penelitian ini akan menetapkan sebuah *website* yang akan dijadikan target serangan. Aplikasi web ini bernama *Mutillidae*. *Mutillidae* adalah aplikasi web yang bersifat *free* dan *open source* yang dirancang dengan memiliki banyak celah keamanan sehingga kita dapat melakukan pengujian serangan.

Pada Gambar 5 memperlihatkan sebuah percobaan *SQL injection* pada aplikasi web *Mutillidae*. Penyerang mencoba melakukan injeksi dengan memasukkan username dan *password* asal' OR '1'='1 pada form login. Pada gambar 6 terlihat bahwa percobaan *SQL injection* tersebut berhasil dan penyerang dapat melakukan login menggunakan akun dengan level admin. Selain *mutillidae* masih banyak aplikasi web lain yang dapat digunakan untuk pembelajaran melakukan *penetrating testing* (Halami, 2010).




**Gambar 5. *Input SQL injection* pada *mutillidae***




**Gambar 6. *SQL injection* pada *mutillidae* berhasil dilakukan**

## Perancangan

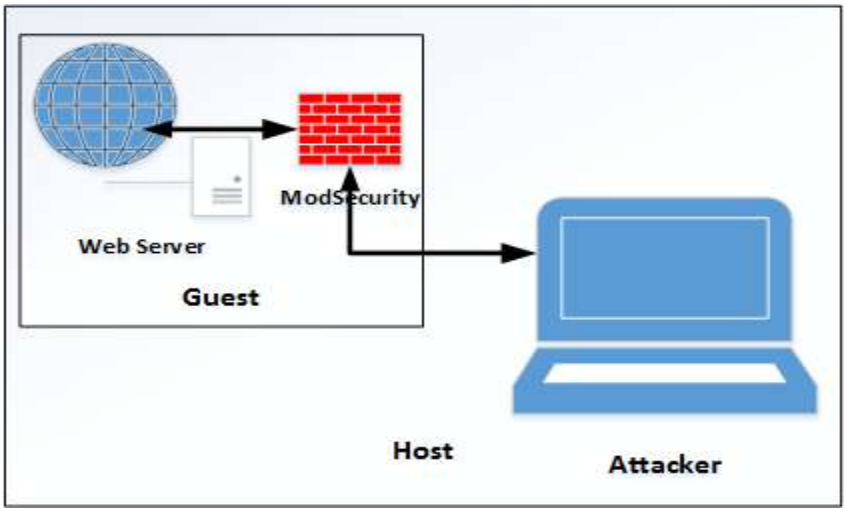
### 1. *Spesifikasi Hardware (host+client)*

Untuk implementasi dan pengujian ini akan dikerjakan pada lingkungan virtual dengan menggunakan aplikasi *virtualbox* dengan spesifikasi sebagai berikut:

i. Mesin target yang juga akan berfungsi sebagai *web server* akan berjalan pada sistem operasi Ubuntu Server 64 bit 12.04 LTS, RAM 1 GB, *Web Server Apache*. Mesin ini akan dibangun sebagai *guest* di *virtualbox.*

ii. Mesin yang digunakan untuk melakukan penyerangan adalah mesin *host* dengan sistem operasi Windows 8 Pro 64 bit.

Lingkungan pengujian penelitian ini dapat dilihat pada gambar 7, di mana arsitektur rancangan *web server* berada di lingkungan *virtualbox.*




**Gambar 7. Arsitektur lingkungan pengujian di *virtualbox***

## 2. Skenario Pengujian

i. Pengujian terhadap serangan *SQL Injection*

Pada pengujian ini akan dilakukan beberapa serangan *SQL injection* untuk mengetahui apakah *ModSecurity* mampu mencegah serangan ini. Beberapa input *SQL injection* yang akan dilakukan adalah sebagai berikut:

- asal' OR '1'='1, akan dilakukan untuk melakukan *bypass* terhadap form login.
- ‘union select 1,2,@@version,4 #, untuk mengetahui versi *MySQL* yang digunakan.
- ‘union select 1,2,@@datadir,4 #, untuk mengetahui direktori database.

Celah keamanan terhadap serangan *SQL Injection* pada input diatas sering terjadi dan memiliki dampak yang cukup parah (Dougherty, 2012).

ii. *Web Load Testing,*

Pengujian ini dilakukan untuk mengetahui kinerja *web server* ketika *ModSecurity* sudah diimplementasikan. Apakah dengan implementasi *ModSecurity* ini kinerja sistem tetap bagus dan *reliable* atau justru menjadi lambat. Parameter yang digunakan untuk pengujian ini adalah response time dari *web server*.

Pada pelaksanaannya nanti pengujian ini akan dilakukan sebanyak 5 kali dari tiap-tiap *request* per *second*. Sedangkan response time akan dicari dari rata-rata ke lima pengujian tersebut untuk masing-masing *request* per *second* yang nilainya 10, 20, 30, 50, 75 dan

100. Pengujian ini akan membandingkan *response time web server* sebelum dan sesudah pemasangan *ModSecurity*.

## B. Implementasi

### 1. Instalasi *Mutillidae*

Aplikasi *Mutillidae* dapat diunduh secara gratis melalui tautan http://sourceforge.net/projects/mutillidae/. Untuk instalasi *Mutillidae* cukup lah mudah, kita cukup melakukan ekstraksi arsip *Mutillidae* ke direktori *web server* kita:

```
fr@websrv:~$ sudo unzip LATEST-mutillidae-2.5.8.zip -d /var/www
```

Kemudian kita rubah konfigurasi database sesuai dengan *Database Server* yang kita punya. Untuk konfigurasi database di *Mutillidae mutillidae* terdapat di file:

/mutillidae/classes/MySQLHandler.php

### 2. Instalasi *ModSecurity*

Untuk mengunduh *ModSecurity* dapat dilakukan menggunakan perintah *wget* yang ada di linux. Versi *ModSecurity* yang ada sekarang adalah versi 2.7.4.

```
fr@websrv:~$ wget –c https://www.modsecurity.org/tarball/2.7.4/modsecurity-apache_2.7.4.tar.gz
--2013-05-31 01:45:10-- https://www.modsecurity.org/tarball/2.7.4/modsecurity-apache_2.7.4.tar.gz
Resolving www.modsecurity.org (www.modsecurity.org)... 204.13.200.240
Connecting to www.modsecurity.org (www.modsecurity.org)|204.13.200.240|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1014983 (991K) [application/x-gzip]
Saving to: `modsecurity-apache_2.7.4.tar.gz'
100%[======================================================>]
1,014,983 49.2K/s in 43s 2013-05-31 01:45:56 (23.0 KB/s) - `modsecurity-apache_2.7.4.tar.gz' saved [1014983/1014983]
```

Setelah mengunduh *ModSecurity*, lakukan esktraksi terhadap paket tersebut.

```
fr@websrv:~$ tar xzvf modsecurity-apache_2.7.4.tar.gz
```

Sebelum melakukan instalasi *ModSecurity* beberapa komponen tambahan yang menjadi *dependencies* untuk paket ini adalah sebagai berikut:

- apxs2

- libxml2
- mod_unique_id

apxs 2 merupakan ekstensi Apache untuk mengkompilasi modul-modul yang akan diintegrasikan dengan Apache. Untuk menginstal ekstensi ini di Ubuntu bisa menggunakan perintah:

```
fr@websrv:~$ sudo apt-get install apache2-prefork-dev
```

libxml2 merupakan pustaka yang digunakan untuk melakukan parsing terhadap XML. Untuk menginstal pustaka ini menggunakan perintah:

```
fr@websrv:~$ sudo apt-get install libxml2
```

Komponen tambahan terakhir yang harus sudah diinstal pada sistem adalah mod_unique_id . uUntuk melakukan instalasi modul ini cukup menambahkan pada httpd.conf baris berikut:

```
fr@websrv:~$ sudo ln -s /etc/apache2/mods-available/unique_id.load /etc/apache2/mods-enabled/
```

Lakukan kompilasi dengan perintah

```
fr@websrv:~/modsecurity-apache_2.7.4$ ./configure
fr@websrv:~/modsecurity-apache_2.7.4$ make
```

**3. *Setting Rules***

Untuk *rule* yang akan diterapkan pada *ModSecurity* ini menggunakan OWASP *ModSecurity Core Rule Project* yang dapat diunduh melalui tautan:

```
https://github.com/SpiderLabs/owasp-modsecurity-crs
```

**4. Instalasi *Httperf***

Untuk pengujian kinerja dari *web server* setelah dipasang *ModSecurity* akan digunakan sebuah aplikasi yang bernama httperf. Aplikasi ini dapat diunduh melalui situs https://code.google.com/p/httperf/. Untuk menjalankan httperf di sistem operasi Windows 8 digunakan Cygwin. Httperf ini akan berjalan diatas Cygwin. Berikut langkah instalasi httperf:

```
$ cd /home/farid/temp
$ tar zxvf httperf-0.9.tar.gz
$ cd httperf-0.8
```

```
$ mkdir build
$ cd build
$ ../configure
$ make
$ make install
```

Kemudian untuk menjalan aplikasi httperf ini dengan menggunakan perintah:

```
$ /usr/local/bin/httperf –-server=192.168.0.103 –-rate=10 –num-conns=1000
```

Di mana *rate* adalah banyaknya *response* per *second*, *server* adalah mesin target dan num-conns adalah banyaknya koneksi yang terjadi.

```
httperf --client=0/1 --server=192.168.0.103 --port=80 --uri=/ --rate=10 --send-buffer=4096
--recv-buffer=16384 --num-conns=1000 --num-calls=1
Maximum connect burst length: 1
Total: connections 1000 requests 1000 replies 1000 test-duration 99.901 s
Connection rate: 10.0 conn/s (99.9 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 1.0 avg 2.6 max 55.7 median 0.5 stddev 4.5
Connection time [ms]: connect 0.9
Connection length [replies/conn]: 1.000
Request rate: 10.0 req/s (99.9 ms/req)
Request size [B]: 66.0
Reply rate [replies/s]: min 9.8 avg 10.0 max 10.2 stddev 0.1 (19 samples)
Reply time [ms]: response 1.7 transfer 0.0
Reply size [B]: header 283.0 content 177.0 footer 0.0 (total 460.0)
Reply status: 1xx=0 2xx=1000 3xx=0 4xx=0 5xx=0
CPU time [s]: user 56.58 system 43.08 (user 56.6% system 43.1% total 99.8%)
Net I/O: 5.1 KB/s (0.0*10^6 bps)
Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
```

## IV. HASIL DAN PEMBAHASAN

### 1. Uji Pencegahan Serangan *SQL injection*

Dari pengujian pencegahan serangan *SQL injection* didapatkan hasil bahwa *ModSecurity* mampu mencegah input *SQL injection* yang ada pada *web target*. *ModSecurity* akan melakukan proteksi terhadap perintah *SQL Injection* yang telah kita set dengan mengenali suatu *request* berdasarkan *reguler expression* yang ada di *core rule*. Apabila suatu *request* yang mengandung perintah *SQL injection* cocok dengan aturan *filter* yang ada di *core rule* maka *request* tersebut akan ditolak atau gagal diteruskan.

**Tabel 1. Tabel hasil pengujian *SQL injection* pada *mutillidae***

| No | Input *SQL Injection* | Sebelum dipasang *ModSecurity* | Sesudah dipasang *ModSecurity* |
|---|---|---|---|
| **1** | asal' OR '1'='1 | berhasil | gagal |
| **2** | ‘union select 1,2,@@version,4 # | berhasil | gagal |
| **3** | ‘union select 1,2,@@datadir,4 # | berhasil | gagal |

**2. Uji *Web Load Testing***

Pada pengujian *load testing* yang dilakukan pada *web server* sebelum dipasang *ModSecurity*, rata-rata *response time web server* pada pengujian *request* per second 10-50 masih terlihat relatif sama. Tetapi setelah *request* per second naik menjadi 75, *response time* meningkat secara signifikan .

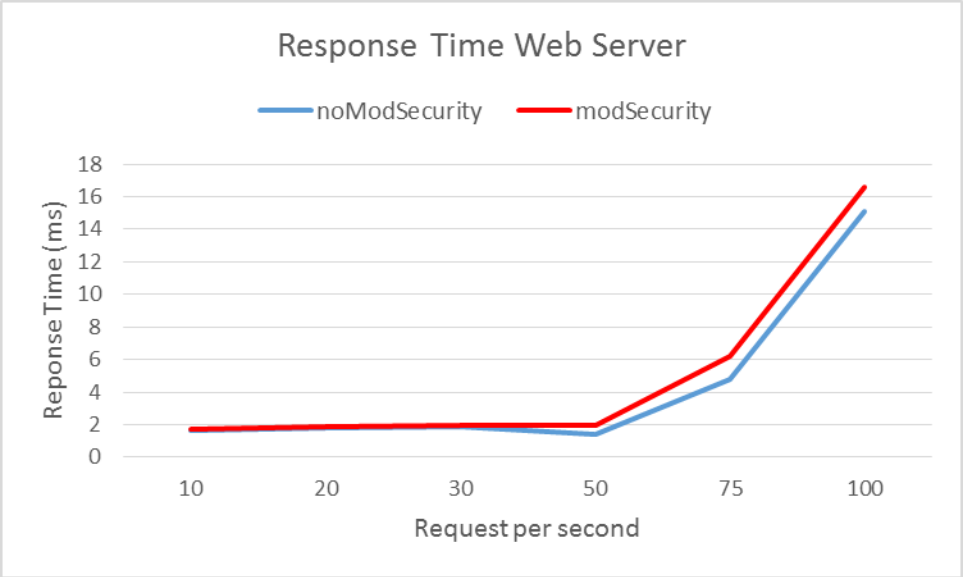
**Tabel 2. *Response time web server* sebelum dipasang *ModSecurity***

| *Req per second* | *Response time* (ms) | | | | | |
|---|---|---|---|---|---|---|
| | **Uji 1** | **Uji 2** | **Uji 3** | **Uji 4** | **Uji 5** | **Rata-rata** |
| **10** | 1.8 | 1.4 | 1.7 | 1.9 | 1.6 | 1.68 |
| **20** | 2.2 | 1.6 | 1.4 | 1.8 | 1.9 | 1.78 |
| **30** | 1.5 | 1.9 | 1.6 | 2.7 | 1.7 | 1.88 |
| **50** | 1.2 | 1.6 | 1.4 | 1.4 | 1.4 | 1.4 |
| **75** | 3.9 | 4.8 | 5.7 | 5 | 4.7 | 4.82 |
| **100** | 18.4 | 10.7 | 9.4 | 14.9 | 21.9 | 15.06 |

Hasil pengujian *load testing* pada *web server* setelah dipasang *ModSecurity* menunjukkan ada peningkatan rata-rata *respons time* dari *Web server*. Untuk mengetahui apakah peningkatan ini signifikan atau tidak akan dilakukan pengujian menggunakan *t-test* menggunakan aplikasi R.

**Tabel 3. *Response time web server* setelah dipasang *ModSecurity***

| *Req per second* | *Response time* | | | | | |
|---|---|---|---|---|---|---|
| | **Uji 1** | **Uji 2** | **Uji 3** | **Uji 4** | **Uji 5** | **Rata-rata** |
| **10** | 1.7 | 1.5 | 1.5 | 1.8 | 2 | 1.7 |
| **20** | 2.2 | 1.5 | 2.3 | 1.7 | 1.6 | 1.86 |
| **30** | 1.6 | 2 | 1.6 | 2.7 | 1.8 | 1.94 |
| **50** | 2.1 | 1.6 | 1.4 | 1.4 | 3.4 | 1.98 |
| **75** | 6.4 | 7 | 7.5 | 5.3 | 4.7 | 6.18 |
| **100** | 19.6 | 11.5 | 11 | 18.8 | 21.9 | 16.56 |




**Gambar 8. Perbandingan *response time* sebelum dan sesudah dipasang *ModSecurity***

$H_0$ = Tidak ada perbedaan *respons time* web server sebelum dan sesudah dipasang *modsecurity* $(\mu_D = 0)$

$H_1$ = Terdapat perbedaan *respons time* Web server sebelum dan sesudah dipasang *modsecurity* $(\mu_D \neq 0)$

x = rata-rata *response time* sebelum dipasang *mod security*

y = rata-rata *response time* setelah dipasang *mod security*

Dari pengujian *t-test* menggunakan aplikasi R didapatkan hasil sebagai berikut:

```
> x = c(1.68, 1.78, 1.88, 1.4, 4.82, 15.06)
> y = c(1.7, 1.86, 1.94, 1.98, 6.18, 16.56)
> t.test(x,y,paired=TRUE)
        Paired t-test
data:  x and y
t = -2.1734, df = 5, p-value = 0.08179
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.3096544  0.1096544
sample estimates:
mean of the differences
                  -0.6
```

Dari hasil pengujian *t-test* di atas nilai p-value 0.08179 lebih besar dari alfa 0.05, dengan demikian H0 diterima yang berarti bahwa performa *web server* tidak terdapat

perbedaan yang signifikan sebelum dipasang *ModSecurity* dan sesudah dipasang *ModSecurity*. Hasil pengujian *load testing* pada *web server* setelah dipasang *ModSecurity* ternyata tidak terlalu berbeda dari sebelumnya. Terdapat peningkatan *response time* yang terjadi tetapi tidak terlalu signifikan . Pola peningkatan lamanya *response time* pun mirip dengan sebelumnya di mana peningkatan secara drastis terlihat ketika *request* per *second* 75 ke atas.

## V. KESIMPULAN DAN SARAN

Dari hasil pengujian di bab sebelumnya dapat disimpulkan bahwa:

i. Implementasi *ModSecurity* sudah berhasil diterapkan.

ii. Penerapan *ModSecurity* dengan menggunakan *core rule* OWASP mampu mencegah terjadinya serangan *SQL injection* pada aplikasi *web mutillidae*.

iii. *ModSecurity* tidak membebani kinerja dari *web server* secara berlebihan sehingga kinerja *ModSecurity* cukup bagus berdasarkan hasil yang didapatkan dari *web load testing*.

*ModSecurity* cukup bagus dalam mencegah serangan *SQL injection*, tetapi untuk melakukan konfigurasi *rule* masih cukup rumit. Untuk penelitian ke depan bisa dikembangkan semacam *graphical user interface* (GUI) yang akan mempermudah kerja admin dalam menentukan *rule* untuk konfigurasi WAF.

## DAFTAR PUSTAKA

Álvarez, G., & Petrovic, S. A. (2003). A Taxonomy of Web Attacks. *Lecture Notes in Computer Science* (hal. 295-298). Berlin: Springer.

Dougherty, C. (2012). *Practical Identification of SQL Injection Vulnerabilities.* Washington, DC: US-CERT.

Halami, W. (2010). *Vulnerable Web Applications for learning.* Diambil kembali dari http://securitythoughts.wordpress.com/2010/03/22/vulnerable-web-applications-for-learning/

Mische, M. (2009). *ModSecurity 2.5 Securing Your Apache Installation and Web Applications.* Brimingham: Packt Publishing Ltd.

Mogul, R., & Lane, A. (2009). *Building a Web Application Security Program.* Phoenix, AZ 85085: Securosis, L.L.C.

OWASP. (2013). *The Ten Most Critical Web Application Security Vulnerabilities*. Diambil kembali dari The Open Web Application Security Project (OWASP): https://www.owasp.org/index.php/Top_10_2013-Top_10

S. Petrushevski, G., & Cabrera, H. (2013). *CloudFlare vs Incapsula vs ModSecurity.* Diambil kembali dari http://zeroscience.mk/files/wafreport2013.pdf

Sumantri, I. (2012, November). *Statistik Insiden Keamanan Internet Indonesia.* Diambil kembali dari Government Computer Security Incident Response Team (GOV-CSIRT): http://govcsirt.kominfo.go.id/download/events/01%20Trend%20Serangan%20Internet-nov2012.pdf

*Web Security Glossary.* (2013). Dipetik Juni 23, 2013, dari Web Application Security Consortium: http://www.webappsec.org/projects/glossary/v1/wasc_glossary_02262004.pdf.

# *USER ACCEPTANCE* TERHADAP SIPADU-STIS MENGGUNAKAN TEORI *TECHNOLOGY ACCEPTANCE MODEL* (*TAM*) DAN METODE ANALISIS *STRUCTURAL EQUATION MODELING* (*SEM*)

**Abialam Koesnandy Hardjantho**
**Staf Badan Pusat Statistik**

**Margaretha Ari Anggorowati**
**Dosen sekolah Tinggi Ilmu statistik**

***Abstract***

*The successful of developing and implementation of information sistem is depend on the user acceptance. SIPADU-STIS is an academic information system that support academic's bisnis process. Technology Acceptance Model (TAM) will be employed for analysing of SIPADU-STIS system. Validation of TAM model will estimate using Structural Equation Model (SEM). The model shows that SIPADU-STIS has a high user's acceptance .*

***Key words** : Technology Acceptance Model (TAM), Structural EquationModeling (SEM), SIPADU-STIS, user acceptance, perceived usefulness, perceived ease of use*

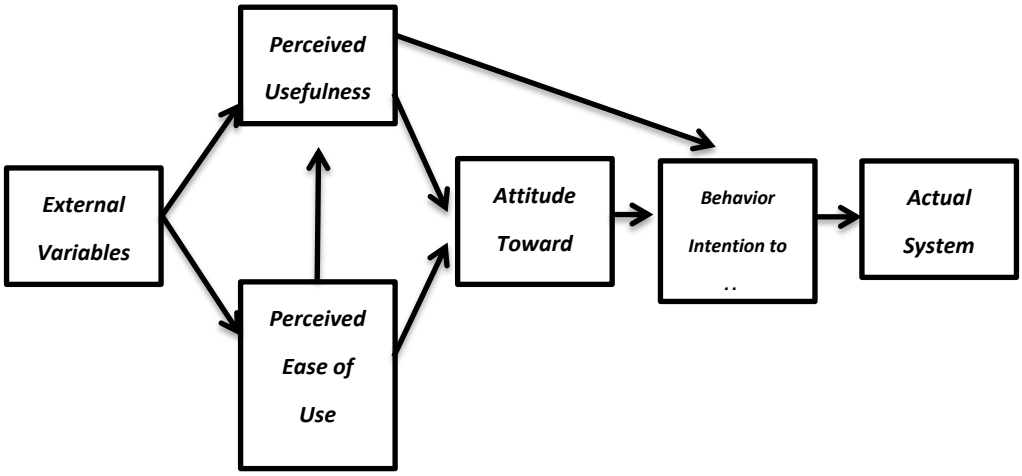
## I. PENDAHULUAN

Sekolah Tinggi Ilmu Statistik (STIS) memiliki sebuah sistem informasi yaitu SIPADU (Sistem Informasi Terpadu) atau yang dikenal dengan SIPADU-STIS. Menurut Kang (1998), penerapan suatu sistem informasi baru dalam suatu organisasi akan memengaruhi keseluruhan organisasi terutama pada sumber daya manusianya. Sehingga kesuksesan dalam pembangunan dan pengembangan sistem informasi sangat bergantung pada tingkat penerimaan pengguna dari system informasi tersebut. Sehingga kesuksesan dalam pembangunan dan pengembangan sistem informasi sangat bergantung pada tingkat penerimaan pengguna dari sistem informasi tersebut. Hal tersebut disebabkan faktor pengguna memiliki pengaruh yang besar dalam menentukan apakah suatu sistem informasi dapat menjalankan tugas yang ditetapkan. Hingga saat ini terdapat beberapa

model yang berkembang untuk menganalisis faktor-faktor yang memengaruhi perilaku pengguna terhadap teknologi seperti *Theory of Reasoned Action* (TRA), *Theory Planned Behavior* (TPB), dan *Technology Acceptance Model* (TAM). Menurut (Chuttur, 2009) TAM banyak digunakan dan dan dikembangkan dalam analisis penerimaan pengguna. Model TAM melakukan pendekatan melalui dua variabel utama yaitu persepsi kemudahan penggunaan (*Perceived ease of use*) dan persepsi kegunaan (*Perceived usefulness*). Analisis model TAM untuk SIPADU-STIS membutuhkan sebuah alat analisis yang akurat. *Structural Equation Modelling* (SEM) digunakan sebagai metode analisis dan validasi model TAM. SEM merupakan suatu teknik *multivariate* yang menggabungkan aspek-aspek regresi berganda analisis jalur dan analisis faktor sehingga dapat memberikan kekuatan estimasi dari semua hipotesis hubungan antar variabel dalam model teoritis dan memberikan informasi dari semua hipotesis pengaruh secara langsung maupun tidak langsung antar satu variabel dengan variabel lainnya.

## II. METODOLOGI

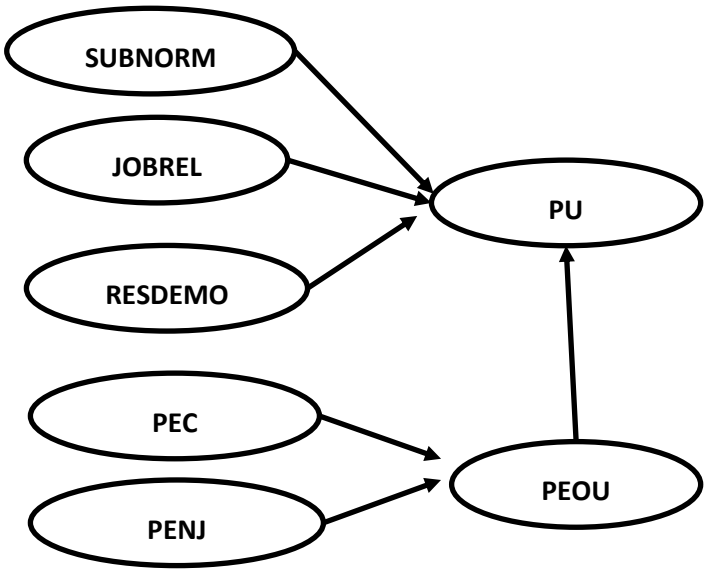
*Technology Acceptance Model* (TAM) merupakan suatu model yang digunakan untuk menjelaskan penerimaan dari sisi pengguna terhadap suatu sistem informasi tertentu. Menurut Davis (1993), TAM memberikan representasi yang informatif dari suatu mekanisme pemilihan desain yang memengaruhi penerimaan pengguna, sehingga dapat diterapkan untuk memprediksi dan mengevaluasi penerimaan pengguna dari suatu teknologi informasi. Model TAM dalam memprediksi dan mengevaluasi penerimaan pengguna berdasarkan pada 2 variabel utama, yaitu persepsi kegunaan (*perceived usefulness*) dan persepsi kemudahan penggunaan (*perceived ease of use*) yang akan memengaruhi niat dan sikap seseorang terhadap penggunaan (*attitude toward using*), hal tersebut kemudian berpengaruh terhadap niat dan perilaku seseorang untuk menggunakan (*behavior of intention to use*) hingga akhirnya menunjukkan pada penggunaan nyata dari suatu sistem tertentu (*actual system use*).




**Gambar 2.** ***Technology Acceptance Model***

Selanjutnya, Venkantesh dan Bala (2008) melakukan penelitian dengan menggabungkan model TAM 2 dan menjelaskan faktor-faktor penentu yang mempengaruhi persepsi kemudahan penggunaan (*Perceived ease of use*). Model TAM 3 berkembang untuk menganalisis pada lingkup yang lebih luas yaitu lingkup organisasi. Berbeda dari TAM 2 yang lebih fokus pada analisis pengguna dalam mengambil keputusan untuk menggunakan atau tidak menggunakan suatu teknologi di tempat kerja. Model TAM 3 memberikan representasi suatu jaringan yang lengkap berkaitan faktor-faktor penentu untuk adaptasi dan penggunaan suatu sistem tertentu.

Pada penelitian ini model TAM disesuaikan pada karakteristik responden, sistem informasi dan organisasi yang diteliti. *Perceived usefulness* (PU) dipengaruhi oleh *subjective norm* (SUBNORM), *job relevance* (JOBREL), *result demonstrability* (RESDEMO), *perceived ease of use* (PEOU). *Perceived ease of use* dipengaruhi oleh *perceptions of external control* (PEC) dan *perceived enjoyment* (PENJ). Gambar 2 menunjukkan kosntruk model TAM yang akan digunakan dalam penelitian ini.




**Gambar 2. Konstruk model TAM**

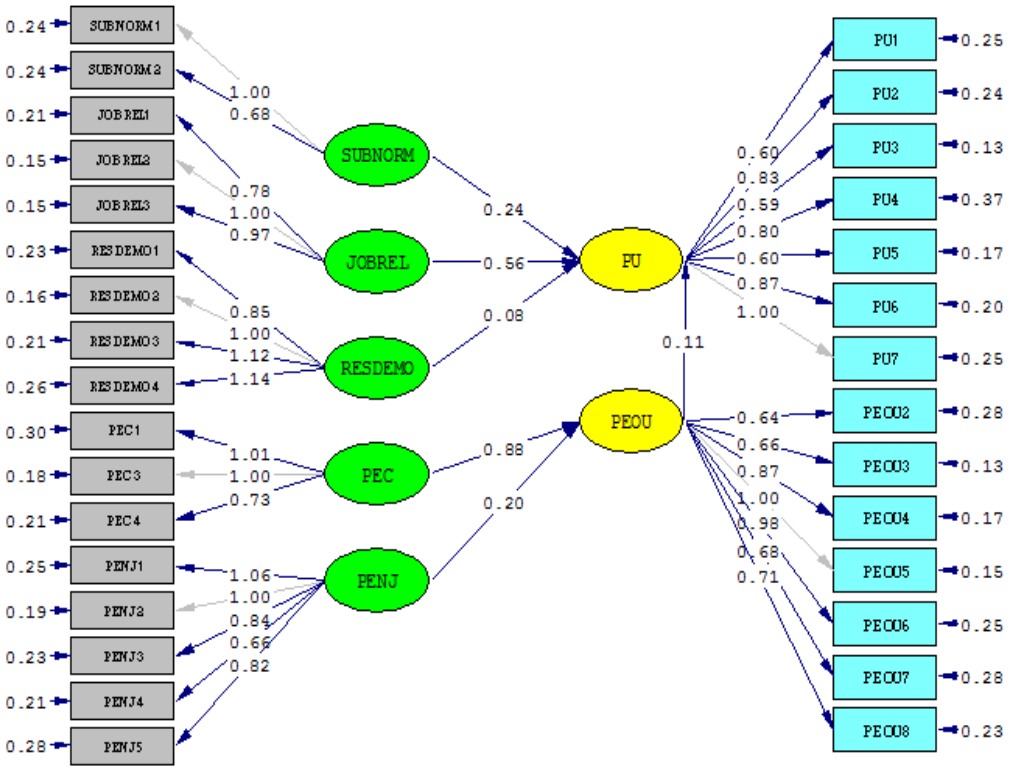
Dalam buku Wijanto (2008), *Structural Equation Modeling* (SEM) adalah suatu teknik statistik yang menganalisis variabel teramati, variabel laten dan error. SEM merupakan suatu teknik analisis yang berupa gabungan dan pengembangan dari beberapa teknik-teknik *multivariate* statistik seperti regresi berganda, analisis faktor dan analisis jalur yang dapat digunakan dalam menganalisis rangkaian variabel yang rumit atau multidimensional.

Pada penelitian yang dilakukan oleh Anggorowati (2013) dengan judul "Pengembangan Metode Estimasi SEM *Non*-Standar Pada Analisis *Technology Acceptance Model*". Hasil pada penelitian ini menunjukkan 3 hubungan signifikan yaitu *management support* kepada *subjective norm*, *subjective norm* kepada *perceived usefulness, perceptions of external control* kepada *perceived ease of use*.

## III. HASIL DAN PEMBAHASAN

### Estimasi Parameter

Pada penelitian ini metode estimasi yang digunakan adalah *Robust Maximum Likelihood.* Hasil estimasi dari berbagai parameter dari model SEM ($\lambda_X$ , $\lambda_Y$, $\varepsilon$ , $\gamma$, $\beta$, $\zeta$) yang disajikan pada Gambar 3.




**Gambar 3. Estimasi parameter Model SEM**

## Uji Kecocokan Model

Dalam menilai *Goodness of fit* antara data dengan model pada metode SEM tidak dapat dilakukan secara langsung dilakukan seperti pada teknik *multivariate* lainnya. SEM tidak memiliki satu uji yang terbaik yang dapat menjelaskan fit atau tidaknya suatu model. Sebagai gantinya, pada metode SEM menggunakan beberapa ukuran *Goodness Of Fit Indices* (GOFI). Pada uji kebaikan model, model SEM memenuhi 10 kriteria baik (*good fit*), 2 kriteria cukup baik (*marginal fit*) dan 3 kriteria kurang baik. Oleh karena itu, dari keseluruhan hasil uji dapat disimpulkan bahwa model yang digunakan merupakan model yang baik (*good fit*). Hail uji kebaikan model ditunjukkan oleh Tabel 1.

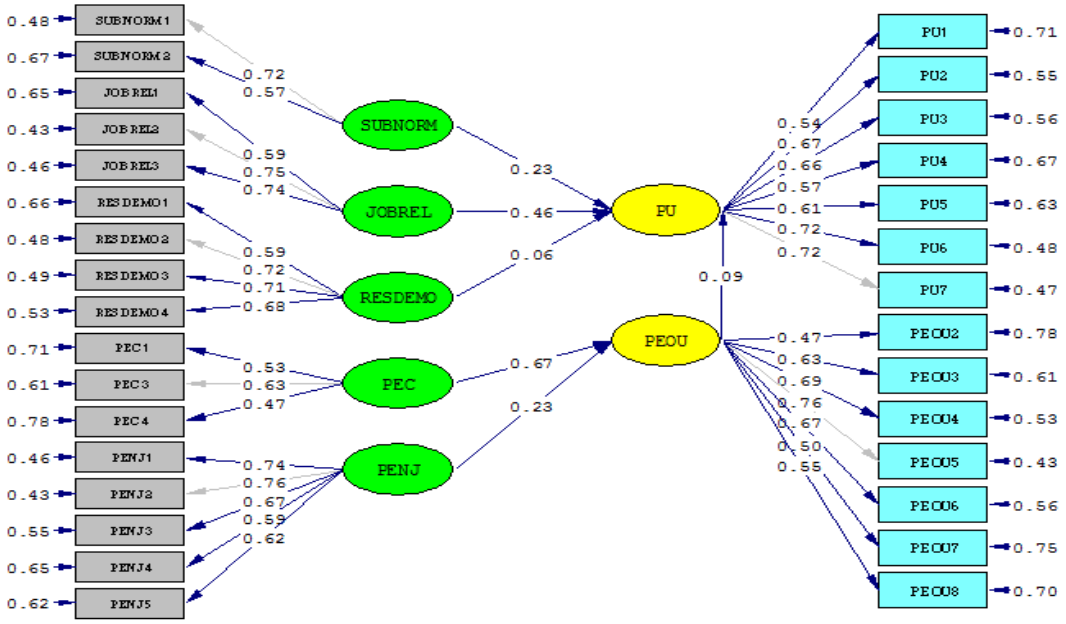
**Tabel 1. Hasil uji kecocokan model ( *goodness of fit* )**

| No | Goodness of Fit | Cut Off | Hasil Estimasi | Tingkat Kecocokan |
|---|---|---|---|---|
| (1) | (2) | (3) | (5) | (4) |
| | | Uji Kecocokan Absolut | | |
| **1** | *Chi-Square* | Nilai yang kecil<br>*p-value* > 0,05 | Chi-square=<br>801,91<br>p-value=0,0000 | Kurang Baik |
| **2** | NCP | Nilai yang kecil<br>Interval yang sempit | NCP : 383,91<br>Interval : (307,67 ;<br>467,94) | Kurang Baik |
| **3** | GFI | GFI ≥ 0,9 | 0,82 | *Marginal Fit* |
| **4** | RMR | *Standardized*<br>RMR ≤ 0,05 | 0,066 | Kurang Baik |
| **5** | RMSEA | RMSA ≤ 0,08 | 0,052 | **Baik** (***good fit***) |
| **6** | ECVI | Nilai yang kecil dan dekat dengan<br>ECVI *saturated* | M* :2,81<br>S* :2,91<br>I* :36,12 | **Baik** (***good fit***) |
| | | Uji Kecocokan Inkremental | | |
| **7** | NNFI | NNFI ≥ 0,9 | 0,96 | **Baik** (***good fit***) |
| **8** | NFI | NFI ≥ 0,9 | 0,93 | **Baik** (***good fit***) |
| **9** | AGFI | AGFI ≥ 0,9 | 0,80 | *Marginal Fit* |
| **10** | RFI | RFI ≥ 0,9 | 0,93 | **Baik** (***good fit***) |
| **11** | IFI | IFI ≥ 0,9 | 0,97 | **Baik** (***good fit***) |
| **12** | CFI | CFI ≥ 0,9 | 0,97 | **Baik** (***good fit***) |
| | | Uji Kecocokan Parsimoni | | |
| **13** | AIC | Nilai yang kecil dan dekat dengan<br>AIC *saturated* | M* : 957,91<br>S* : 992,00<br>I* : 12317,96 | **Baik** (***good fit***) |
| **14** | CAIC | Nilai yang kecil dan dekat dengan<br>CAIC *saturated* | M* : 1335,02<br>S* : 3390,07<br>I* : 12467,84 | **Baik** (***good fit***) |
| | | Ukuran Kecocokan Hoetler's Critical N | | |
| **15** | CN | CN ≥ 200 | 208,60 | **Baik** (***good fit***) |

## Analisis Model Pengukuran

Tahap selanjutnya yang harus dilakukan setelah pengujian kecocokan model adalah analisis model pengukuran. Analisis model pengukuran meliputi evaluasi terhadap validitas variabel teramati (indikator) dan evaluasi reliabilitas dari model pengukuran setiap variabel laten.

Setelah melakukan pengolahan dengan menggunakan program LISREL 8.8 berupa model SEM SLF dan *t-value*, dapat diperoleh informasi beberapa indikator yang dianggap telah valid sehingga dilakukan eliminasi untuk indikator tersebut.




**Gambar 3. Model SEM: nilai *standardized loading factor***

## Interpretasi Model Pengukuran

Pada Gambar 3 dapat kita peroleh informasi berupa nilai muatan faktor standar (*standardized loading factor*) yang menunjukkan seberapa besar kontribusi variabel teramati dalam pembentukan variabel latennya. Variabel teramati yang memiliki nilai *standardized loading factor* terbesar memberikan informasi bahwa variabel teramati tersebut dapat dengan baik menggambarkan variabel laten yang di ukurnya. Selanjutnya nilai *variance extracted* yang dihasilkan menunjukkan seberapa besar informasi dalam variabel teramati dapat diwakili oleh variabel latennya.

## Analisis Model Struktural

Analisis model struktural merupakan analisis yang dilakukan untuk mengevaluasi parameter-parameter yang menunjukkan hubungan kausal atau pengaruh antar variabel laten dalam penelitian. Berdasarkan diagram jalur yang dihasilkan, dapat diperoleh pengaruh antar variabel laten. Model struktural yang dihasilkan adalah :

PU = 0,11*PEOU + 0,24*SUBNORM + 0,56*JOBREL + 0,082*RESDEMO,

PEOU = 0,88*PEC + 0,20*PENJ, Errorvar.= 0,064

Signifikansi koefisien jalur pada model struktural ditunjukkan oleh Tabel 2.

**Tabel 2. Evaluasi signifikansi koefisien pada model struktural**

| Jalur | Estimasi | *t-value* | Kesimpulan |
|---|---|---|---|
| **(1)** | (2) | (3) | (5) |
| **SUBNORM → PU** | 0,24 | 2,00 | Signifikan |
| **JOBREL → PU** | 0,56 | 4,69 | Signifikan |
| **RESDEMO → PU** | 0,082 | 0,68 | Tidak Signifikan |
| **PEOU → PU** | 0,11 | 0,95 | Tidak Signifikan |
| **PEC → PEOU** | 0,88 | 4,35 | Signifikan |
| **PENJ → PEOU** | 0,20 | 2,08 | Signifikan |

**a) Hipotesis 1: *Subjective norm* diduga memiliki pengaruh terhadap *perceived usefulness***

Koefisien jalur antara variabel *subjective Norm* dengan *perceived usefulness* yaitu 0,24 dan nilai *t-value* sebesar 2,00 lebih besar dari nilai t-tabel pada taraf signifikan α =0,05 yaitu 1,96 yang menyatakan *subjective norm* yang dimiliki atau dirasakan berpengaruh signifikan terhadap *perceived usefulness* pengguna SIPADU-STIS. Sedangkan nilai positif pada koefisien parameter menunjukkan semakin tingginya tingkat *subjective norm* yang dimiliki atau dirasakan maka akan semakin tinggi tingkat *perceived usefulness* pengguna SIPADU-STIS.

**b) Hipotesis 2: *Job relevance* diduga memiliki pengaruh terhadap *perceived usefulness***

Koefisien jalur antara variabel *job relevance* dengan *perceived usefulness* yaitu 0,56 dan nilai *t-value* sebesar 4,69 lebih besar dari nilai t-tabel pada taraf signifikan α =0,05 yaitu 1,96 yang menyatakan *job relevance* yang dimiliki atau dirasakan berpengaruh signifikan terhadap *perceived usefulness* pengguna SIPADU-STIS. Sedangkan nilai positif pada koefisien parameter menunjukkan semakin tingginya tingkat *job relevance* yang dimiliki atau dirasakan maka akan semakin tinggi tingkat *perceived usefulness* pengguna SIPADU-STIS.

**c) Hipotesis 3: *Result demonstrability* diduga memiliki pengaruh terhadap *perceived usefulness***

Koefisien jalur antara variabel *result demonstrability* dengan *perceived usefulness* yaitu 0,082 dan nilai statistik tabel sebesar 0,68 lebih kecil dari nilai t-tabel pada taraf signifikan α =0,05 yaitu 1,96 yang menyatakan *result demonstrability* yang dimiliki atau dirasakan tidak berpengaruh signifikan terhadap *perceived usefulness* pengguna SIPADU-STIS. Sedangkan nilai positif pada koefisien parameter menunjukkan semakin tingginya tingkat *result demonstrability* yang dimiliki atau dirasakan maka akan semakin tinggi tingkat *perceived usefulness* pengguna SIPADU-STIS.

**d) Hipotesis 4: *Perceived ease of use* diduga memiliki pengaruh terhadap *perceived usefulness***

Koefisien jalur antara variabel *perceived ease of use* dengan *perceived usefulness* yaitu 0,11 dan nilai *t-value* sebesar 0,95 lebih kecil dari nilai t-tabel pada taraf signifikan α =0,05 yaitu 1,96 yang menyatakan *perceived ease of use* yang dimiliki atau dirasakan berpengaruh signifikan terhadap *perceived usefulness* pengguna SIPADU-STIS. Sedangkan nilai positif pada koefisien parameter menunjukkan semakin tingginya tingkat *perceived ease of use* yang dimiliki atau dirasakan maka akan semakin tinggi tingkat *perceived usefulness* pengguna SIPADU-STIS.

**e) Hipotesis 5: *Perceptions of external control* diduga memiliki pengaruh terhadap *perceived ease of use***

Koefisien jalur antara variabel *perceptions of external control* dengan *perceived ease of use* yaitu 0,88 dan nilai *t-value* sebesar 4,35 lebih besar dari nilai t-tabel pada taraf signifikan α =0,05 yaitu 1,96 yang menyatakan *perceptions of external control* yang dimiliki atau dirasakan berpengaruh signifikan terhadap *perceived ease of use* pengguna SIPADU-STIS. Sedangkan nilai positif pada koefisien parameter menunjukkan semakin tingginya tingkat *perceptions of external control* yang dimiliki atau dirasakan maka akan semakin tinggi tingkat *perceived ease of use* pengguna SIPADU-STIS.

**f) Hipotesis 6:** ***Perceived enjoyment diduga memiliki pengaruh terhadap perceived ease of use***

Koefisien jalur antara variabel *perceived enjoyment* dengan *perceived ease of use* yaitu 0,20 dan nilai *t-value* sebesar 2,08 lebih besar dari nilai t-tabel pada taraf signifikan $\alpha$ =0,05 yaitu 1,96 yang menyatakan *perceived enjoyment* yang dimiliki atau dirasakan berpengaruh signifikan terhadap *perceived ease of use* pengguna SIPADU-STIS. Sedangkan nilai positif pada koefisien parameter menunjukkan semakin tingginya tingkat *perceived enjoyment* yang dimiliki atau dirasakan maka akan semakin tinggi tingkat *perceived ease of use* pengguna SIPADU-STIS.

## V. KESIMPULAN DAN SARAN

Pada studi kasus sistem SIPADU, variabel *result demonstrability* tidak berpengaruh signifikan terhadap persepsi kegunaan (PU) dan variabel persepsi kemudahan (PEOU) tidak berpengaruh signifikan terhadap persepsi kegunaan (PU).

Variabel job relevance memiliki pengaruh lebih kuat pada persepsi kemudahan dibandingkan dengan variabel *subjective norm* dan PEOU dan *result demonstrability*, dan variabel *perception of external control* memiliki pengaruh yang lebih kuat pada PEOU dibandingkan dengan variabel *percieved enjoyment*.

Perlu dilakukan pengembangan SIPADU lebih lanjut khususnya ketersediaan sumberdaya yang dibutuhkan dalam penggunaan SIPADU. Kenyamanan pengguna dalam menggunakan SIPADU menjadi faktor yang cukup berperan dalam persepsi kemudahan bagi pengguna SIPADU. Fungsi-fungsi pada SIPADU sebaiknya selalu dikembangkan sesuai dengan tugas pokok para pengguna, sehingga dengan fungsi SIPADU yang terus berkembang akan selalu mendukung proses kerja dari pengguna (*job relevance*).

## DAFTAR PUSTAKA

Arikunto, Suharsimi. (2006). *Prosedur Penelitian Suatu Pendekatan Praktik*. Jakarta: Rineka Cipta.

Anggorowati, M.A. (2013). *Pengembangan Metode Estimasi SEM Non-Standar Pada Analisis Technology Acceptance Model* [Disertasi]. Surabaya: Institut Teknologi Sepuluh November

Azwar, Saifuddin. (1997). *Reliabilitas dan Validitas*. Yogyakarta: Pustaka Pelajar.

Chuttur. M.Y. (2009). Overview Of The Technology Acceptance Model: Origins, Developments And Future Directions. Indiana University, USA. *Sprouts:Working paper on information systems*.

Cochran, G., William. (1991). *Teknik Penarikan Sampel Jilid III.* Jakarta: UI Press.

Davis, F. (1989), Perceived Usefulness, Perceived Ease Of Use And User Acceptance Of Information Technology. *MIS Quartely,* Vol.13 (3),hal. 319-339.

Davis, F. (1993). User Acceptance Of Information Technology: System Characteristics, User Perceptions. *Int. J. Man Machine Studies,* Vol.38 (3),hal. 475-87.

Diponegoro, Ahmad Muhammad. (2005). Validitas Konstruk Skala Afek. *Humanitas: Indonesian Psychological Journal*, 2 No.1 Januari 2005: 64-74.

Djaali, & Pudji. (2008). *Pengukuran dalam Bidang Pendidikan.* Jakarta: Grasindo.

Ghozali, Imam. (2008). *Structural Equation Modeling Teori Konsep dan Aplikasi dengan Program Lisrel 8.80 + CD.* Semarang: Universitas Diponegoro.

Hair, J.F. Jr., Anderson, R.E., Tatham, R.L., & Black, W.C. (1998). *MultivariateData Analysis, (5th Edition).* Upper Saddle River, NJ: Prentice Hall.

Hair, J.F. Jr.,Babin, B.J., Anderson, R.E., & Black, W.C. (2010). *Multivariate Data Analysis, (7th Edition).* Prentice Hall.

Jöreskog, K. G., & Sörbom, D. (1996). *LISREL 8 user's reference guide*. Uppsala, Sweden: Scientific Software International.

Kang, Sungmin. (1998). Information Technology Acceptance : Evolving With The Changes In The Network Environment Center For Information

System Management Department Of Management Science And Information System Graduate School Of Business. *The University of Texas at Austin. IEEE*. 118

Maruyama, G., (1997). Basics Of Structural Equation Modeling. 1st Edn., *Sage Publications, Thousand Oaks*, ISBN-10: 0803974086, pp: 311.

Mike, Rosebush. (2011). Validition of the Character Mosaic Report. *Technical Report.*

Singarimbun, Masri & Sofyan Effendi. (1989). *Metode Penelitian Survei*, LP3ES. Jakarta.

Sugiyono. (2008). *Metode Penelitian Bisnis*. Bandung: Alfabeta.

Takdir. (2011). SIPADU STIS versi Juli 2011. 12 Juli 2014.

Venkatesh, V., & Davis, F.D., (2000). A Theoretical Extension Of The Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*. Vol. 46, No. 2, pp. 186-204.

Venkatesh, V. & H. Bala, (2008). Technology Acceptance Model 3 And A Research Agenda On Interventions. *Decision Sci.,* 39: 273-315.

Venkatesh, V., & Michael G. Moris, (2000), Why on‟t Men Ever Stop to Ask for Directions? Gender, Social Influence, and Their Role in Technology Acceptance and Usage Behavior,*MIS Quarterly*.

Wibisono, Dermawan. (2003). *Riset Bisnis: Panduan bagi Praktisi dan Akademisi*. Jakarta: Gramedia Pustaka Utama.

Wijanto, Setyo Hari. (2008). *Structural Equation Modeling dengan Lisrel 8.8: Konsep dan Tutorial*. Jakarta: Graha Ilmu.