BAB II

KAJIAN PUSTAKA

2.1. Teori Dasar

Berikut ini peneliti akan membahas mengenai teori dasar terkait penelitian menggunakan beberapa kutipan yang dikutip oleh peneliti baik dari buku maupun jurnal penelitian sebelumnya:

2.1.1. Jaringan Komputer

Menurut (Maslan & Wangdra, 2012: 2), jaringan komputer adalah sebuah kumpulan dari komputer, printer, dan peralatan lainnya yang terhubung dalam satu kesatuan dan membentuk suatu sistem tertentu. Komputer-komputer yang terhubung ini dapat saling berkomunikasi, bertukar informasi, maupun berbagi pakai sumber daya yang sama, seperti printer, scanner, perangkat lunak dan lain sebagainya. Komputer-komputer dapat terhubung baik itu melalui media transmisi berupa kabel maupun nirkabel.

Sedangkan menurut (Nugroho, 2016: 9), jaringan merupakan sebuah konsep hubungan/interkoneksi antar sekumpulan perangkat. Antar perangkat harus saling terhubung, apabila ada perangkat yang tidak terhubung, maka konsep tersebut bukan termasuk dalam definisi jaringan.

2.1.2. Standar Jaringan Komputer

Agar komunikasi data dapat terjadi, terlebih dahulu suatu jaringan harus dapat berkomunikasi dengan jaringan yang lain. Oleh karena itu, suatu jaringan komputer yang dibangun harus memperhatikan arsitektur standar yang telah ditentukan. Dengan demikian, perbedaan jenis komputer ataupun sistem operasi tidak menjadi masalah karena jaringan tersebut menggunakan protokol yang sama dalam pengiriman maupun penerimaan data. Salah satu protokol yang pada umumnya digunakan untuk komunikasi data pada jaringan adalah protokol TCP/IP. Menurut (Maslan & Wangdra, 2012: 47), protokol TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data komputer di Internet. Sedangkan menurut (Siregar, 2010: 14–15), protokol TCP/IP adalah kumpulan protokol yang digunakan untuk menghubungkan komputer-komputer yang ada di seluruh dunia. Dengan adanya protokol tersebut, komputer-komputer yang berbeda jenis maupun sistem operasi tetap dapat berkomunikasi.

2.1.3. Jenis Jaringan Komputer

Jenis jaringan komputer dapat dibagi berdasarkan kriteria seperti sebagai berikut:

2.1.3.1. Jenis Jaringan Komputer Berdasarkan Jangkauan

Berdasarkan jangkauannya, jaringan komputer terbagi menjadi 3 jenis:

2.1.3.1.1. LAN (Local Area Network)

Menurut (Andi, 2010: 2), LAN merupakan jaringan yang menghubungkan sejumlah komputer yang ada dalam suatu lokasi dengan area yang terbatas seperti ruang atau gedung. Umumnya LAN memilki cakupan sampai dengan beberapa kilometer.

2.1.3.1.2. MAN (Metropolitan Area Network)

Menurut (Andi, 2010: 3), MAN merupakan jaringan yang lebih besar dari jaringan LAN tetapi lebih kecil dari jaringan WAN. Jaringan MAN dapat dikatakan sebagai gabungan dari beberapa jaringan LAN. Cakupan dari MAN dapat mencakup sampai dengan luas suatu kota.

2.1.3.1.3. WAN (Wide Area Network)

Menurut (Andi, 2010: 2), WAN merupakan jaringan antara LAN satu dengan LAN lain yang dipisahkan oleh lokasi yang cukup jauh. Contoh dari WAN adalah jaringan yang menghubungkan antara suatu kantor pusat dengan kantor cabang di daerah-daerah.

2.1.3.2. Jenis Jaringan Komputer Berdasarkan Media Koneksi

Berdasarkan media koneksinya, jaringan komputer juga terbagi menjadi 3 jenis:

2.1.3.2.1. Fiber Optik

Menurut (Winarno & Zaki, 2013: 65), fiber optik merupakan sebuah teknologi komunikasi yang menggunakan benang kaca atau plastik untuk mentransmisikan data. Pentransmisian dilakukan dengan data dimodulasikan terlebih dahulu sebagai gelombang cahaya. Kabel fiber optik memiliki beberapa keunggulan seperti sebagai berikut:

1. Memiliki bandwidth yang lebih besar dibandingkan dengan kabel tembaga atau kabel logam lainnya.
2. Memiliki ketahanan yang lebih tinggi terhadap interferensi dibandingkan dengan kabel lainnya.
3. Secara fisik memiliki ukuran yang lebih kecil dibandingkan dengan kabel lainnya.
4. Dengan kabel fiber optik, data bisa ditransmisikan secara digital dan bukan analog.

Meskipun begitu, kabel fiber optik juga memiliki beberapa kelemahan, seperti biaya yang lebih mahal baik kabelnya maupun biaya instalasinya, selain itu kabel fiber juga lebih rapuh dan sulit dipotong.

2.1.3.2.2. Ethernet

Menurut (Winarno & Zaki, 2013: 66), ethernet merupakan arsitektur LAN yang dikembangkan oleh Xerox bekerja sama dengan DEC dan Intel di tahun 1976. Ethernet merupakan standar utama yang digunakan dalam membangun

jaringan kabel. Contoh yang termasuk dalam ethernet adalah kabel UTP (*Unshielded Twisted Pair*).

2.1.3.2.3. Wireless LAN

Menurut (Winarno & Zaki, 2013: 67), wireless LAN adalah sebuah jaringan dimana tidak ada koneksi fisik yang menghubungkan antar komputer di jaringan. Komputer-komputer tersebut dihubungkan menggunakan gelombang radio atau gelombang mikro agar dapat berkomunikasi satu sama lain.

2.1.4. Model OSI Layer

Menurut (Maslan & Wangdra, 2012: 37), OSI (*The Open System Interconnection*) adalah suatu standar komunikasi antarmesin yang terdiri atas 7 lapisan. Sedangkan menurut (Siregar, 2010: 11), OSI merupakan acuan untuk *network communication*. Model OSI dikatakan *open systems architecture* karena model ini menghubungkan satu komputer dengan komputer yang lain menggunakan komunikasi terbuka. OSI dibuat oleh lembaga ISO (*The International Organization for Standardization*), Amerika Serikat. Ketujuh lapisan tersebut mempunyai peran dan fungsi yang berbeda antara satu sama lain. Ketujuh lapisan tersebut dibagi lagi dalam 2 tingkatan grup, yaitu: *upper layer* dan *lower layer*. *Upper layer* terdiri dari *Application Layer, Presentation Layer, Session Layer* dan berfokus pada aplikasi pengguna serta representasi file di komputer, sedangkan *lower layer* terdiri dari *Transport Layer, Network Layer, Data Link Layer, Physical Layer* dan berfokus pada *network engineering* yang

membuat perangkat keras. Berikut ini akan dijelaskan satu per satu fungsi dari masing-masing layer:

2.1.4.1. Physical Layer

Menurut (Sugeng, 2010: 66), *Physical layer* berfungsi dalam pengiriman raw bit ke kanal komunikasi. Masalah-masalah yang harus diperhatikan adalah masalah desain (jika dikirim bit 1 harus diartikan bit 1 di sisi penerima), masalah desain ini ditemukan ada hubungannya dengan mekanika, kelistrikan, prosedur *interface*, dan medium transmisi fisik yang berada di bawah lapisan fisik. *Physical layer* juga merupakan lapisan yang paling dekat dengan perangkat keras jaringan secara fisik. Perangkat yang beroperasi pada lapisan ini adalah *hub*, *repeater*, *network interface card*, dan *host bus adapter*.

2.1.4.2. Data Link Layer

Sedangkan *data link layer* menurut (Sugeng, 2010: 66–67), tugas utamanya sebagai fasilitas transmisi raw data dan mentransformasikan data tersebut ke saluran yang bebas dari kesalahan transmisi. Pada lapisan ini dimungkinkan melakukan pemecahan data input menjadi sejumlah *frame*. Selanjutnya *frame* tersebut dikirim secara berurutan, dan memproses *acknowledgement frame* yang dikirim kembali oleh penerima. Perangkat yang beroperasi pada lapisan ini adalah *bridge* dan *switch layer-2*.

2.1.4.3. Network Layer

N*etwork layer* menurut (Sugeng, 2010: 67), berfungsi sebagai pengendalian operasi subnet. Masalah desain yang penting adalah menentukan route pengiriman paket dari sumber ke tujuannya. Lapisan ini akan mendefinisikan alamat IP (*Internet Protocol*) perangkat sehingga setiap perangkat dapat berkomunikasi dalam satu jaringan. Selain itu lapisan ini juga berfungsi dalam membuat *header* pada paket data dan kemudian menentukan jalur pengiriman yang akan ditempuh oleh paket data. Perangkat yang beroperasi pada lapisan ini adalah *router* dan *switch layer-3.*

2.1.4.4. Transport Layer

T*ransport layer* menurut (Sugeng, 2010: 67), memiliki fungsi dasar menerima data dari *session layer*, bila perlu memecah data menjadi bagian-bagian yang lebih kecil, meneruskan potongan data ke lapisan jaringan dan menjamin seluruh potongan data sampai dengan benar di sisi lainnya. Lapisan ini bertanggungjawab dalam memastikan semua paket akan tiba pada penerima tanpa ada *error*. Apabila ada paket yang hilang atau rusak dalam proses pengiriman, *transport layer* akan mengirim ulang paket tersebut.

2.1.4.5. Session Layer

Menurut (Sugeng, 2010: 68), *session layer* yang bertugas mengizinkan para pengguna untuk menetapkan *session* diantara mereka dan *session* tersebut memungkinkan seorang pengguna untuk melakukan log ke dalam suatu *remote*

time sharing system atau memindahkan suatu file dari satu mesin ke mesin yang lain. Dengan kata lain, lapisan ini akan mendefinisikan bagaimana sebuah *session* atau koneksi akan dibangun, dan lapisan ini akan melakukan kontrol terhadap koneksi yang dibangun seperti menghancurkan maupun memelihara koneksi yang telah ada.

2.1.4.6. Presentation Layer

Presentation layer menurut (Sugeng, 2010: 68), melakukan fungsi-fungsi tertentu yang sering diminta untuk menjamin penemuan sebuah penyelesaian umum bagi masalah tertentu. *Presentation layer* juga bertugas mentranslasikan data yang ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan dan juga sebaliknya. Contoh protokol yang berada pada lapisan ini adalah *Remote Desktop Protocol (RDP)*.

2.1.4.7. Application Layer

Berbanding terbalik dengan *physical layer*, *application layer* merupakan lapisan yang paling dekat dengan *user*. Menurut (Sugeng, 2010: 68), lapisan ini berfungsi melayani *remote terminal*. Lapisan ini terdiri dari berbagai macam protokol yang biasa dipergunakan. Lapisan ini juga bertugas menyediakan antarmuka ke jaringan melalui aplikasi untuk digunakan *user* serta mengatur bagaimana aplikasi dapat mengakses jaringan dan membuat pesan-pesan kesalahan. Contoh protokol yang berada pada lapisan ini adalah HTTP, FTP, POP3.

2.2. Teori Khusus

Berikut ini peneliti akan membahas mengenai teori khusus terkait penelitian menggunakan beberapa kutipan yang dikutip oleh peneliti baik dari buku maupun jurnal penelitian sebelumnya:

2.2.1. Web Server

Menurut (Laksana & Rosyid, 2017: 364), *web server* merupakan komputer yang memiliki tanggung jawab untuk menerima HTTP *request* dari *client* dan mengirimkan HTTP *response* menuju *client*. HTTP *request* yang dimaksud adalah pesan yang dikirimkan dari *client* ke *server*, termasuk baris pertama dari pesan tersebut, metode yang digunakan, penanda dari sumber dan versi protokol yang digunakan. HTTP *request* umumnya dikirim oleh *client* menggunakan *web browser*.

2.2.2. SQL Injection

Pada subbab berikut, peneliti akan melakukan pembahasan lebih lanjut mengenai teknik SQL Injection:

2.2.2.1. Definisi SQL Injection

Menurut (Baihaqi et al., 2013: 2), SQL injection adalah sebuah serangan dengan cara memasukkan sebuah kode ke dalam sebuah string yang akan

dieksekusi oleh aplikasi *database server*. Sedangkan menurut (Efendi, Yovita, & Hafidudin, 2016: 279), SQL Injection adalah sebuah metode menyisipkan *query* sql ke dalam sebuah aplikasi *web* melalui form *input* dan url (*uniform resource locator*) untuk mendapatkan informasi pada basis data.

Pada umumnya, *database* akan langsung melakukan eksekusi *query* dengan nilai input yang diterima dari *website* tanpa melakukan pemeriksaan terhadap nilai input terlebih dahulu. Padahal, nilai input tertentu dapat mengubah arti atau tujuan dari query SQL yang sebenarnya. SQL Injection bekerja dengan mengeksploitasi kelemahan tersebut. Contohnya query SQL di bawah ini:

$txId = ($_POST[‘User_Id’]);

$txSQL = "SELECT * FROM Member WHERE User_Id = ‘”.$txId.”’

“;

Katakanlah seorang penyerang melakukan input pada parameter “User_Id” dengan nilai **31 OR 1=1**. Kemudian *website* menerima input tersebut dan mengeksekusi *query* tanpa melakukan validasi apapun terhadap nilai input, sehingga query SQL akan menjadi:

SELECT * FROM Member WHERE User_Id = 31 OR 1=1;

Query SQL di atas valid sehingga database akan memberi balasan dengan memberikan seluruh baris dari tabel “Member” karena OR 1=1 selalu benar. Dengan demikian, penyerang telah memperoleh informasi dari database tersebut. Padahal, tujuan sebenarnya dari *query* SQL tersebut adalah untuk mencari informasi dari 1 user saja.

2.2.2.2. Jenis-Jenis SQL Injection

Pada umumnya, SQL Injection terdiri dari 3 jenis, yaitu:

2.2.2.2.1. Union-Based SQL Injection

Jenis ini merupakan jenis SQL Injection yang sering digunakan. Menurut (Baloch, 2015: 343), *this type of attack utilizes the use of a UNION statement, which is the combination of two select statements, to extract information from the database.* Jenis ini memanfaatkan perintah "UNION" untuk menggabungkan dua perintah pada SQL dan mengambil informasi dari *database*. Contohnya adalah: **http://localhost/index.php?support=yes' and 1=0 UNION all select 1,2,3,4--±**.

Penyisipan perintah di atas dapat menampilkan informasi dari kolom *database*. Penggunaan "1=0" berfungsi agar hasil dari *query* yang terletak di sebelah kiri "UNION" tidak dieksekusi, sedangkan *query* "all select 1,2,3,4--±" merupakan perintah yang menampilkan informasi dari kolom *database*. Penggunaan UNION di sini memungkinkan penambahan *query* "all select 1,2,3,4--±".

2.2.2.2.2. Error-Based SQL Injection

Jenis SQL Injection ini merupakan jenis SQL Injection yang paling mudah digunakan. Menurut (Baloch, 2015: 343), i*n this technique, we cause an application to throw an error to extract the database*. Dengan menggunakan teknik ini, ketika seorang penyerang menyisipkan perintah SQL pada input, *database* akan membalas *request* tersebut dalam bentuk pesan *error* yang

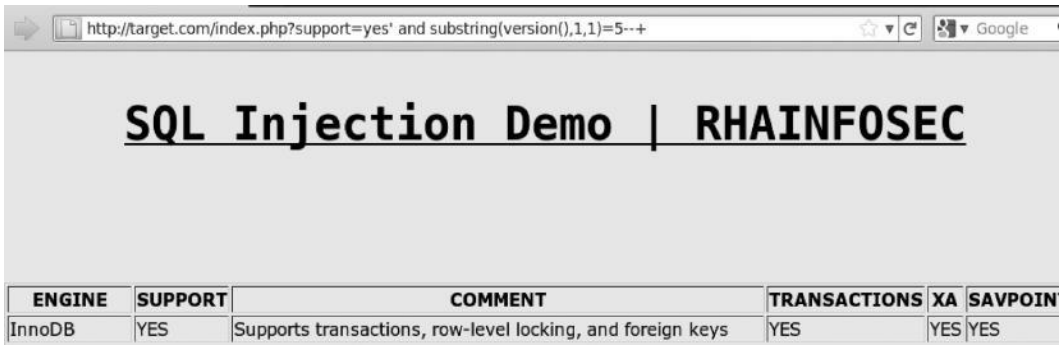
ditampilkan langsung pada *website*. Dari *error* tersebut, penyerang mendapatkan informasi dari *database*. Contohnya: **http://www.example.com/product.aspx?Id=7 and @@version=1--**. Penyisipan *query* "**and @@version=1—**" pada *website* tertentu, dapat menampilkan informasi mengenai jenis *database* serta sistem operasi yang sedang digunakan.

2.2.2.2.3. Blind SQL Injection

Teknik ini biasanya digunakan apabila *error-based* SQL injection tidak berhasil oleh karena *website* telah diatur untuk tidak menampilkan pesan *error*. Menurut (Baloch, 2015: 355), *a blind SQL injection is one where an attacker extracts the data by asking the database "true or false" questions*. Pada blind SQL injection, untuk mendapatkan informasi mengenai *database*, penyerang perlu melakukan injeksi SQL dalam bentuk pertanyaan "benar atau salah". Penyerang kemudian akan menarik kesimpulan berdasarkan *output* dari *website* karena tidak adanya pesan *error* yang ditampilkan. Injeksi perlu dilakukan terus menerus sampai penyerang mendapatkan informasi yang diinginkannya. Oleh karena itu, teknik ini juga merupakan jenis SQL Injection yang paling sulit digunakan. Contohnya: seorang penyerang ingin menggunakan teknik Blind SQL Injection untuk mengetahui versi *database* MySQL yang digunakan.

Gambar 2.1 False result hasil dari Blind SQL Injection
Sumber: Buku Ethical Hacking and Penetration Testing Guide, p. 358

Query "http://localhost/index.php?support=yes' AND SUBSTRING(version),1,1)=**4**;--+" digunakan untuk memastikan apakah versi mysql yang digunakan adalah versi 4 dan hasilnya *website* memberikan *false result* seperti gambar di atas yang berarti versi MySQL yang digunakan bukan versi 4. Sedangkan ketika digunakan *query* "http://localhost/index.php?support=yes' AND SUBSTRING(version (),1,1)=**5**;--+", *website* memberikan *output* seperti gambar di bawah.



Gambar 2.2 True result hasil dari Blind SQL Injection
Sumber: Buku Ethical Hacking and Penetration Testing Guide, p. 358

Output yang diberikan merupakan *true result*, hal ini menunjukkan bahwa versi MySQL yang digunakan adalah versi 5. Hasilnya, penyerang tetap dapat mengali informasi mengenai *database* tanpa menggunakan pesan *error* meskipun prosesnya lebih sulit.

2.2.3. Honeypot

Dalam subbab berikut, peneliti akan melakukan pembahasan lebih lanjut mengenai Honeypot:

2.2.3.1. Definisi Honeypot

Menurut (Baihaqi et al., 2013: 1) , Honeypot adalah sebuah sistem yang dibuat untuk menyimulasikan layanan yang berjalan di atas sebuah server dengan tujuan sebagai umpan untuk mengalihkan atau menarik perhatian peretas untuk menyerang sistem tersebut. Sedangkan menurut (Aidin et al., 2016: 2173), Honeypot adalah suatu sistem yang sengaja dikorbankan untuk menjadi sasaran penyerangan dari peretas. Honeypot bekerja dengan mengelabui peretas dan mengalihkan peretas dari sistem yang lain. Dengan demikian, Honeypot dapat melindungi sistem yang lebih vital.

Selain mengalihkan serangan, Honeypot juga dapat diimplementasikan untuk mencatat informasi mengenai aktivitas peretasan yang dilakukan oleh seorang peretas. Aktivitas ini kemudian dapat dipelajari untuk keperluan pengembangan sistem yang lain agar tingkat keamanan sistem tersebut dapat meningkat dan peretasan dengan teknik yang sama dapat dihindari.

2.2.3.2. Jenis-Jenis Honeypot

Berdasarkan tingkat interaksi yang dapat dilakukan suatu Honeypot dengan penyerang, Honeypot dibagi menjadi 2, yaitu:

2.2.3.2.1. Low Interaction Honeypot

Menurut (Kaur, 2012: 3), *low interaction honeypot have limited interaction; they normally work by emulating services and operating systems.* Hal ini memungkinkan Honeypot diimplementasikan dengan menggunakan sumber daya yang lebih sedikit karena memiliki tingkat interaksi yang terbatas dengan penyerang. Akan tetapi, karena adanya keterbatasan interaksi, informasi aktivitas peretasan yang dapat dicatat dari penyerang kemungkinan terbatas karena sistem tidak mendukung penuh teknik penyerangan yang digunakan peretas. Selain itu karena keterbatasan interaksi, penyerang juga dapat lebih mudah menyadari bahwa sistem yang diretas merupakan suatu sistem Honeypot. Meskipun begitu, Honeypot jenis ini memiliki kelebihan, yaitu lebih cepat, mudah diimplementasikan dan lebih aman jika dibandingkan Honeypot jenis yang lain.

2.2.3.2.2. High Interaction Honeypot

Kebalikan dari *low interaction honeypot*, *high interaction honeypot* memilki tingkat interaksi yang tinggi dengan penyerang. Menurut (Kaur, 2012: 3), *high interaction honeypot are usually complex solutions as they involve real operating systems and applications*. Hal ini berarti Honeypot tidak hanya melakukan tiruan dari layanan, fungsi, maupun sistem operasi saja. Honeypot jenis ini memberikan sistem dan layanan nyata layaknya sistem yang sesungguhnya. Karena

memberikan sistem secara penuh, jenis Honeypot ini memiliki resiko yang lebih tinggi untuk dimanfaatkan peretas dalam menyerang sistem yang lain sehingga membutuhkan *maintenance* dan *monitoring* lebih lanjut. Meskipun demikian, informasi aktivitas peretasan yang dapat dicatat oleh Honeypot juga lebih tinggi.

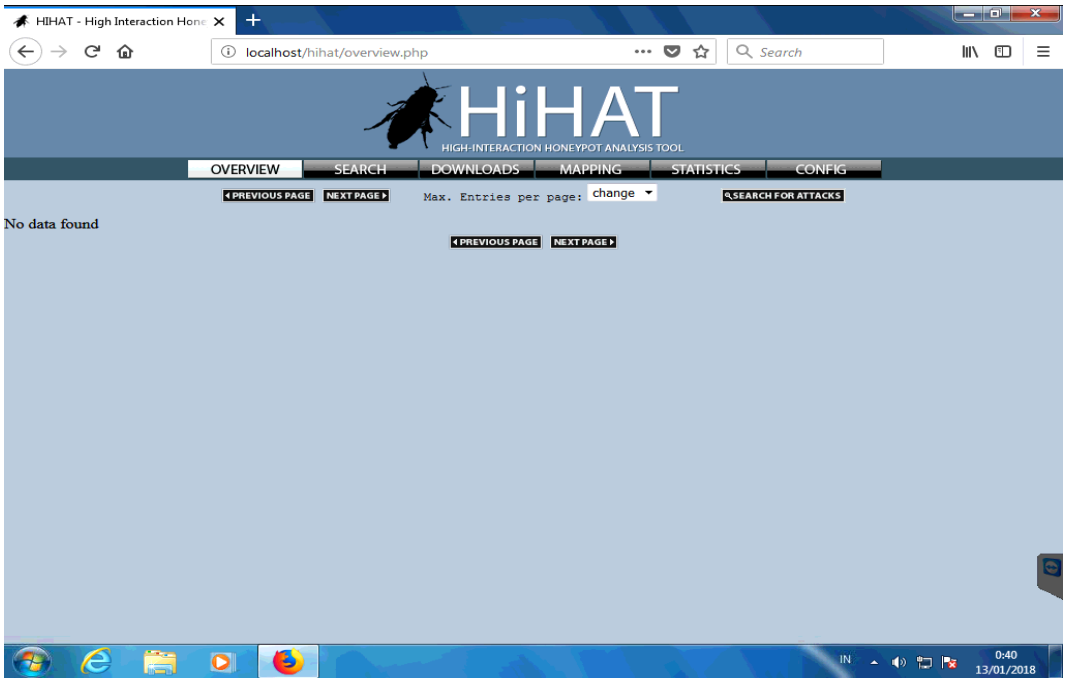
2.3. Tools

Peralatan pendukung yang akan digunakan oleh peneliti dalam penelitian ini antara lain:

1. High Interaction Honeypot Analysis Toolkit (HIHAT)

Menurut (Nugraha, Djanali, & Pratomo, 2013: 2), *High Interaction Honeypot Analysis Toolkit* (HIHAT) adalah sebuah perangkat lunak yang mampu mengubah aplikasi PHP menjadi sebuah *high interaction honeypot*. HIHAT bekerja dengan menambahkan perintah-perintah PHP ke semua file halaman *website*. Dengan perintah tersebut, HIHAT dapat mendeteksi serangan, serta mencatat semua *request* yang masuk ke dalam sistem Honeypot dan menyimpannya ke dalam sebuah *database*. Kemudian HIHAT akan menyediakan suatu antarmuka sederhana dalam bentuk *website* yang dapat digunakan untuk memantau maupun menganalisis informasi yang berhasil dicatat. Penambahan perintah PHP ke file halaman *website*, tidak akan mengubah tampilan atau fungsi dari halaman *website* tersebut. Oleh karena itu, peretas tidak dapat mengetahui bahwa halaman *website* yang sedang diakses merupakan halaman *website* palsu yang telah disediakan oleh HIHAT. Ketika seorang peretas mengirimkan request

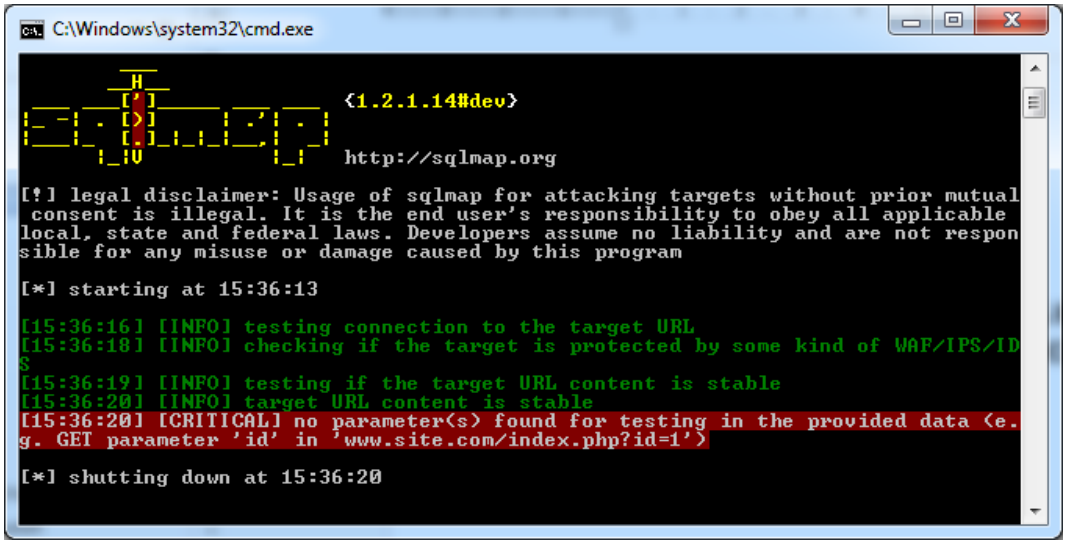
yang berbahaya, Honeypot akan memproses dan membalas *request* tersebut sama halnya dengan *website* yang asli namun dengan informasi yang berbeda.



Gambar 2.3 HIHAT
Sumber: Peneliti

2. Sqlmap

Sqlmap merupakan *penetration testing tool* yang bersifat *open-source* yang dapat secara otomatis melakukan SQL Injection pada *website*. Sqlmap menggunakan CLI (*command-line interface*) sebagai antarmukanya. Tool ini juga dapat digunakan pada beberapa sistem operasi, seperti: Windows, Linux/UNIX, Mac OS X, Solaris. Sqlmap ditulis dengan bahasa pemrograman Python. Oleh karena itu sebelum digunakan, Python harus ter-*install* pada komputer. Versi Python yang mendukung sqlmap adalah versi 2.6.x dan 2.7.x.



Gambar 2.4 Sqlmap
Sumber: Peneliti

2.4. Penelitian Terdahulu

Berikut ini terdapat beberapa penelitian terdahulu yang relevan dengan penelitian yang peneliti lakukan, meskipun terdapat sedikit perbedaan dengan penelitian peneliti:

1. Penelitian yang dilakukan oleh Sainath Patil, Nageshri B Karhade, dan Yogini K Kothekar pada tahun 2012. Judul penelitiannya adalah "***Honeyweb: a web-based high interaction client honeypot***". Pada penelitian ini, peneliti menggunakan Honeypot untuk mempelajari serangan-serangan yang dilakukan pada website. Peneliti mengubah suatu web PHP menjadi Honeypot. Hasilnya, Honeypot berhasil mendeteksi serangan-serangan, termasuk SQL Injection dan mencatat *request* tersebut.

Berdasarkan penelitian ini, peneliti mengambil kesimpulan bahwa Honeypot dapat digunakan sebagai sistem untuk mendeteksi serangan SQL Injection.

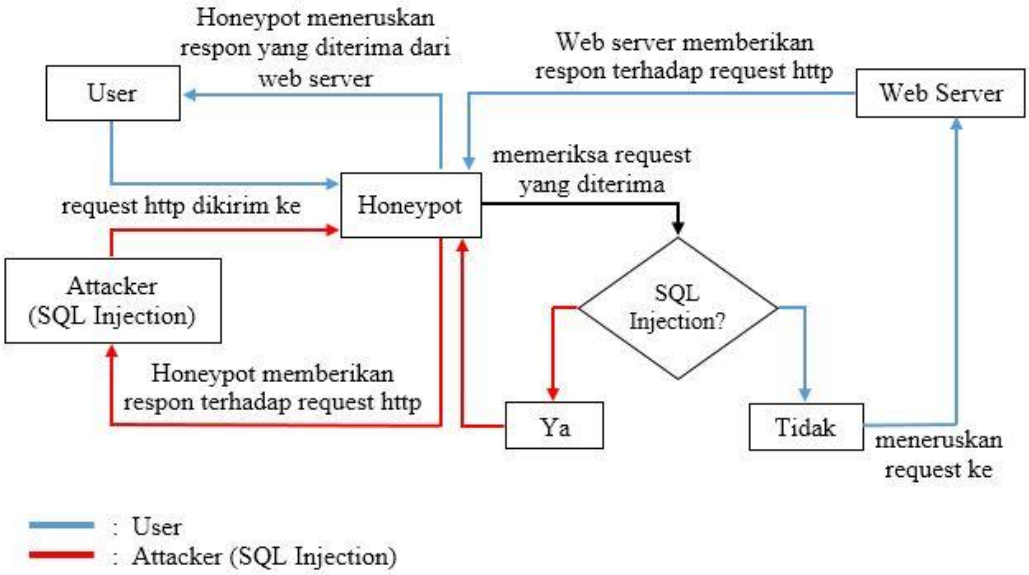
2. Penelitian yang dilakukan oleh Satrio Gita Nugraha, Supeno Djanali, dan Baskoro Adi Pratomo pada tahun 2013. Judul penelitiannya adalah **"Sistem Pendeteksi dan Pencegah Serangan SQL Injection dengan Penghapusan Nilai Atribut Query SQL dan Honeypot"**. Pada penelitian ini, peneliti menggunakan Honeypot untuk melindungi *web server* dari peretas. Peneliti menggunakan suatu proxy untuk mendeteksi SQL Injection dan proxy tersebut akan memutuskan apakah *request* diteruskan ke *web server* atau ke Honeypot. Untuk mendeteksi SQL Injection, proxy tersebut menggunakan perintah-perintah PHP yang ditelah ditentukan. Hasilnya, sistem berhasil mendeteksi serangan dan mengalihkannya ke Honeypot. Berdasarkan penelitian ini, peneliti mengambil kesimpulan bahwa Honeypot dapat mencegah peretasan *website* dengan teknik SQL Injection.
3. Penelitian yang dilakukan oleh Abdurrazak Baihaqi, Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., dan Baskoro Adi Pratomo, S.Kom., M.Kom. pada tahun 2013. Judul penelitiannya adalah **"Perancangan Web Application Honeypot untuk Menggali Informasi Peretas"**. Pada penelitian ini, peneliti menggunakan Honeypot untuk mengalihkan serangan peretas dari *web server*. Berdasarkan hasil uji coba, Honeypot dapat menjalankan perannya dengan baik dalam menyimulasikan *output* bagi serangan SQL injection. Berdasarkan penelitian ini, peneliti mengambil

kesimpulan bahwa Honeypot mencegah serangan SQL Injection dengan menirukan atau mensimulasikan kinerja dari suatu sistem yang lindungi.

4. Penelitian yang dilakukan oleh Lukito Prima Aidin, Surya Michrandi Nasution, dan Fairuz Azmi pada tahun 2016. Judul penelitiannya adalah **"IMPLEMENTASI HIGH INTERACTION HONEYPOT PADA SERVER"**. Pada penelitian ini, peneliti menggunakan Honeypot untuk mengalihkan serangan peretas dari *web server*. Setelah implementasi Honeypot selesai, Honeypot diuji menggunakan berbagai metode dan salah satunya adalah SQL Injection. Peneliti menggunakan tools pada kali Linux untuk melancarkan serangan pada Honeypot. Hasilnya Honeypot berhasil mengemulasikan dan mencatat semua serangan, kecuali DoS. Berdasarkan penelitian ini, peneliti mengambil kesimpulan bahwa peretasan pada website dengan teknik SQL Injection dapat dicegah dengan Honeypot.
5. Penelitian yang dilakukan oleh Ms. Khairkar Ashwini, Gollar Pratiksha, Kulakarni Anuja, Suryawanshi Varsharani, dan Swami Gayatri pada tahun 2017. Judul penelitannya adalah **"Secure Network System using Honeypot"**. Pada penelitian ini, peneliti menggunakan Honeypot untuk melindungi suatu *server*. Untuk dapat berkomunikasi dengan *server* dalam jaringan, setiap *request* harus melalui Honeypot terlebih dahulu. Apabila *request* yang dikirimkan berasal dari *user normal*, maka request akan diteruskan ke *server* dan *server* akan membalas *request* tersebut melalui Honeypot. Sebaliknya apabila yang mengirimkan *request* adalah attacker, maka Honeypot sendiri yang akan memberikan respon kepada *request*

tersebut dan Honeypot akan mencatat informasi mengenai *request* tersebut. Berdasarkan penelitian ini, peneliti mengambil kesimpulan bahwa Honeypot dapat mencegah peretasan yang dilakukan dengan teknik SQL Injection.

2.5. Kerangka Pemikiran



Gambar 2.5 Kerangka Pemikiran
Sumber: Peneliti

Setiap *request http* yang dikirimkan pada *website* akan diterima dan diperiksa terlebih dahulu oleh Honeypot. Apabila Honeypot menilai *request* yang diterima tidak tergolong SQL Injection, maka Honeypot akan meneruskan *request* tersebut ke *web server*. Kemudian *web server* akan memberikan respon terhadap *request* tersebut kembali kepada Honeypot dan Honeypot akan meneruskan lagi respon yang didapatkan dari *web server* kepada *user*. Namun apabila Honeypot

menemukan bahwa *request* tersebut tergolong SQL Injection, maka *request* tidak akan diteruskan ke *web server*. Sebaliknya, Honeypot akan membalas *request* tersebut seolah-olah sebagai *web server* untuk mengelabui peretas dan melindungi *web server*. Honeypot juga akan mencatat segala aktivitas yang terjadi pada proses tersebut.