ANALISIS SERANGAN SQL INJECTION PADA WEB SERVER MENGGUNAKAN INTRUSION DETECTION SYSTEM

Disusun dan Diajukan untuk Memenuhi Salah Satu Syarat Guna Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA	**: PRENDI PARLUHUTAN.S**
N.P.M	**: 1614370320**
PROGRAM STUDI	**: SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

LEMBAR PENGESAHAN

ANALISIS SERANGAN SQL INJECTION PADA WEB SERVER MENGGUNAKAN INTRUSION DETECTION SYSTEM

Disusun Oleh :

Nama : **PRENDI PARLUHUTAN.S**

NPM : **1614370320**

Program Studi : **SISTEM KOMPUTER**

Skripsi telah disetujui oleh Dosen Pembimbing Skripsi

Pada Tanggal

Dosen Pembimbing I

Dian Kurnia, S.Kom.,M.Kom

Dosen Pembimbing II

Ika Devi Perwitasari S.Kom.,M.Kom

Mengetahui,

Dekan Fakultas Sains dan Teknologi

Hamdani S.T.,MT

Ketua Program Sistem Komputer

Eko Hariyanto S.Kom.,M.Kom

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini :

Nama : Prendi Parluhutan Simandalahi

NPM : 1614370320

Prodi : Sistem Komputer

Konsentrasi : Keamanan Jaringan Komputer

Judul Skripsi : Analisis Serangan SQL Injection Pada Web Server Menggunakan Intrusion Detection System

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya bukan hasil plagiat.
2. Saya tidak akan menuntut perbaikan nilai Indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau.
3. Skripsi saya dapat dipublikasikan oleh lembaga, dan saya tidak akan menuntut akibat publikasi tersebut.

Demikian pernyataan ini saya perbuat dengan sebenar-benarnya, terimakasih.

Medan, 22 Agustus 2020

Yang membuat pernyataan

Prendi Parluhutan Simandalahi

SURAT PERNYATAAN

ya Yang Bertanda Tangan Dibawah Ini :

ama	:	PRENDI PARLUHUTAN SIMANDALAHI
P. M	:	1614370320
mpat/Tgl. Lahir	:	MEDAN / 23 AGUSTUS 1998
amat	:	JL.BAHAGIA LK VI NO 2-D
. HP	:	082165857748
ma Orang Tua	:	S.SIMANDALAHI/H.TAMBA
kultas	:	SAINS & TEKNOLOGI
ogram Studi	:	Sistem Komputer
dul	:	ANALISIS SERANGAN SQL INJECTION PADA WEB SERVER MENGGUNAKAN INTRUSION DETECTION SYSTEM

nsama dengan surat ini menyatakan dengan sebenar - benarnya bahwa data yang tertera diatas adalah sudah benar sesuai ngan ijazah pada pendidikan terakhir yang saya jalani. Maka dengan ini saya tidak akan melakukan penuntutan kepada MPAB. Apabila ada kesalahan data pada ijazah saya.

mikianlah surat pernyataan ini saya buat dengan sebenar - benarnya, tanpa ada paksaan dari pihak manapun dan dibuat lam keadaan sadar. Jika terjadi kesalahan, Maka saya bersedia bertanggung jawab atas kelalaian saya.

Medan, 01 Juli 2020

yataan

PRENDI PARLUHUTAN SIMANDALAHI
1614370320

FM-BPAA-2012-041

Hal : Permohonan Meja Hijau

Medan, 15 Agustus 2020
Kepada Yth : Bapak/Ibu Dekan
Fakultas SAINS & TEKNOLOGI
UNPAB Medan
Di -
Tempat

Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama	: PRENDI PARLUHUTAN SIMANDALAHI
Tempat/Tgl. Lahir	: MEDAN / 23 AGUSTUS 1998
Nama Orang Tua	: S.SIMANDALAHI
N. P. M.	: 1614370320
Fakultas	: SAINS & TEKNOLOGI
Program Studi	: Sistem Komputer
No. HP	: 082165857748
Alamat	: JL.BAHAGIA LK VI NO 2-D

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul **ANALISIS SERANGAN SQL INJECTION PADA WEB SERVER MENGGUNAKAN INTRUSION DETECTION SYSTEM**, Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
3. Telah bercap keterangan bebas pustaka
4. Terlampir surat keterangan bebas laboratorium
5. Terlampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
6. Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkipnya sebanyak 1 lembar.
7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah dijilid lux 2 examplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 examplar untuk penguji (bentuk dan warna penjilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan perincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	0
2. [170] Administrasi Wisuda	: Rp.	1,500,000
3. [202] Bebas Pustaka	: Rp.	100,000
4. [221] Bebas LAB	: Rp.	5,000
Total Biaya	: Rp.	**1,605,000**

Peripde Wisuda Ke : 65

Ukuran Toga : L

Diketahui/Disetujui oleh :

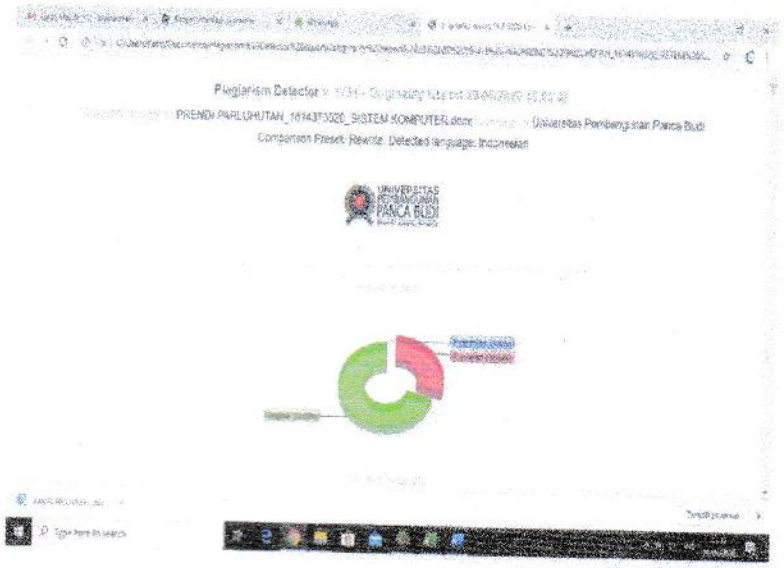
Hamdani, ST., MT
Dekan Fakultas SAINS & TEKNOLOGI

Hormat saya

PRENDI PARLUHUTAN SIMANDALAHI
1614370320

Catatan :

- 1.Surat permohonan ini sah dan berlaku bila ;
  - a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
  - b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
- 2.Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asli) - Mhs.ybs.



SURAT KETERANGAN PLAGIAT CHECKER

Dengan ini saya Ka.LPMU UNPAB menerangkan bahwa saurat ini adalah bukti pengesahan dari LPMU sebagi pengesah proses plagiat checker Tugas Akhir/ Skripsi/Tesis selama masa pandemi ***Covid-19*** sesuai dengan edaran rektor Nomor : 7594/13/R/2020 Tentang Pemberitahuan Perpanjangan PBM Online.

Demikian disampaikan.

NB: Segala penyalahgunaan/pelanggaran atas surat ini akan di proses sesuai ketentuan yang berlaku UNPAB.

Ka.LPMU

Cahyo Pramono, SE.,MM

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITASI)
PROGRAM STUDI ARSITEKTUR	(TERAKREDITASI)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITASI)
PROGRAM STUDI PETERNAKAN	(TERAKREDITASI)

PERMOHONAN JUDUL TESIS / SKRIPSI / TUGAS AKHIR*

ang bertanda tangan di bawah ini :

Lengkap	: prendi parluhutan.s
t/Tgl. Lahir	: MEDAN / 23 Agustus 1998
Pokok Mahasiswa	: 1614370320
n Studi	: Sistem Komputer
trasi	: Keamanan Jaringan Komputer
Kredit yang telah dicapai	: 141 SKS, IPK 3.37
Hp	: 082165857748
ini mengajukan judul sesuai bidang ilmu sebagai	:

Judul

NALISIS SERANGAN SQL INJECTION PADA WEB SERVER MENGGUNAKAN INTRUSION DETECTION SYSTEM0

isi Oleh Dosen Jika Ada Perubahan Judul

g Tidak Perlu

Medan, 18 Mei 2020

Rektor I,

(Cahyo Pamungkas, SE., MM)

Pemohon,

(Prendi Parluhutan.s)

Tanggal : .....

Disahkan oleh :

Dekan

(Hamdani, ST., MT)

Tanggal : .....

Disetujui oleh :

Dosen Pembimbing I :

(Dian Kurnia, S.Kom., M.Kom)

Tanggal : .....

Disetujui oleh:

Ka. Prodi Sistem Komputer

Tanggal : .....

Disetujui oleh:

FM-BPAA-2012-038

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
Medan - Indonesia

Universitas	: Universitas Pembangunan Panca Budi
Fakultas	: SAINS & TEKNOLOGI
Dosen Pembimbing I	: Dian kurnia, S.kom., M.kom
Dosen Pembimbing II	: Ika Devi Perwitasari S.kom, M.kom
Nama Mahasiswa	: PRENDI PARLUHUTAN.S
Jurusan/Program Studi	: Sistem Komputer
Nomor Pokok Mahasiswa	: 1614370320
Jenjang Pendidikan	: Strata 1
Judul Tugas Akhir/Skripsi	: Analisis Serangan SQL Injection Pada Web Server menggunakan Intrusion Detection System

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
25/11-2019	Perbaiki Bab 1		
28/11-2019	Acc Bab 1, lengkapi berkas seminar proposal		Acc seminar proposal
03/12-2019	Perbaiki Bab 2		
15/01-2020	Acc Bab 1 lanjut Bab 3		
04/02-2020	Acc Bab 2 lanjut Bab 3		
16/03-2020	perbaiki bab 3 tahapan pendek		
19/03-2020	Acc bab 3 lanjut Bab 4		

Medan, 22 November 2019
Diketahui/Disetujui oleh :
Dekan,

Sri Shindi Indira, S.T.,M.Sc.

YAYASAN PROF. DR. H. KADIRUN YAHYA

UNIVERSITAS PEMBANGUNAN PANCA BUDI

JL. Jend. Gatot Subroto KM 4,5 PO. BOX 1099 Telp. 061-30106057 Fax. (061) 4514808
MEDAN - INDONESIA
Website : www.pancabudi.ac.id - Email : admin@pancabudi.ac.id

LEMBAR BUKTI BIMBINGAN SKRIPSI

Nama Mahasiswa : PRENDI PARLUHUTAN SIMANDALAHI
NPM : 1614370320
Program Studi : Sistem Komputer
Jenjang Pendidikan : Strata Satu
Dosen Pembimbing : Dian Kurnia, S.Kom., M.Kom
Judul Skripsi : ANALISIS SERANGAN SQL INJECTION PADA WEB SERVER MENGGUNAKAN INTRUSION DETECTION SYSTEM0

Tanggal	Pembahasan Materi	Status	Keterangan
08 April 2020	ACC Bab 1, 2 & 3 Lanjutkan penulisa skripsi kamu ke Bab 4	Disetujui	
17 April 2020	Perbaiki susunan bab 4 kamu sesuai dengan alur tahapan implementasi apliaksi dan pembahasan, sudah saya kirim langusng ke WA kamu, reviewe saya yang detail, silahkan cek	Revisi	
20 April 2020	ACC BAB 4 Lanjut pengerjaan ke Bab 5 kamu	Disetujui	
25 April 2020	Revisi Bab 5 kamu, perjelas kesimpulan kamu sesauikan dengan rumusan masalah kamu , untuk saran buat yang singkat, jangan teori kamu perjelas di saran.	Revisi	
27 April 2020	Lengkapi berkas seminar hasil, dari cover sampai dengan lampiran	Revisi	
24 Juni 2020	Lengkapi berkas Seminar Meja Hijau, ACC Sidang	Disetujui	
06 Agustus 2020	Lengkapi berkas jilid, ACC JILID	Disetujui	
10 Agustus 2020	Lengkapi berkas ACC jilid lux	Disetujui	

Medan, 20 Agustus 2020
Dosen Pembimbing,

Dian Kurnia, S.Kom., M.Kom

FM-BPAA-2012-0

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi
Fakultas : SAINS & TEKNOLOGI
Dosen Pembimbing I : Dian kurnia S.kom., M.kom
Dosen Pembimbing II : Ika Dewi Parwitasari, S.kmo., M.kom
Nama Mahasiswa : PRENDI PARLUHUTAN.S
Jurusan/Program Studi : Sistem Komputer
Nomor Pokok Mahasiswa : 1614370320
Jenjang Pendidikan : Strata 1
Judul Tugas Akhir/Skripsi : Analisis Serangan SQL Injection Pada Web Server Menggunakan Intrusion Detection System

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
25/11 2019	Perbaiki penulisan bab I lihat pandoan		Acc Seminar
12/03 2020	Acc bab I Perbaiki bab 2 & daftar pustaka		
13/03 2020	Acc bab II Perbanyak spasi daftar pustaka		

Medan, 25 November 2019
Diketahui/Disetujui oleh :
Dekan,

YAYASAN PROF. DR. H. KADIRUN YAHYA

UNIVERSITAS PEMBANGUNAN PANCA BUDI

JL. Jend. Gatot Subroto KM 4,5 PO. BOX 1099 Telp. 061-30106057 Fax. (061) 4514808
MEDAN - INDONESIA
Website : www.pancabudi.ac.id - Email : admin@pancabudi.ac.id

LEMBAR BUKTI BIMBINGAN SKRIPSI

Nama Mahasiswa	:	**PRENDI PARLUHUTAN SIMANDALAHI**
NPM	:	**1614370320**
Program Studi	:	**Sistem Komputer**
Jenjang Pendidikan	:	**Strata Satu**
Dosen Pembimbing	:	**Ika Devi Perwitasari, S.Kom., M.Kom**
Judul Skripsi	:	**ANALISIS SERANGAN SQL INJECTION PADA WEB SERVER MENGGUNAKAN INTRUSION DETECTION SYSTEM0**

Tanggal	Pembahasan Materi	Status	Keterangan
14 April 2020	Revisi bab 3 penulisan judul tabel dan spasi isi tabel	Revisi	
4 Mei 2020	Bab 3: Halaman 30 cek kelengkapan huruf, jgn disingkat, cek utk keseluruhan halaman. Halaman 31 cek tata bahasa penggunaan imbuhan di-, cek utk keseluruhan halaman. Judul keterangan tabel rata kiri.	Revisi	
4 Mei 2020	Bab 4: Hal 27, gunakn hirarki penomoran poin spt yg sudh saya jelaskan. Cek lagi beberapa istilah asing belum dimiringkan. Hal 43, perbaiki kata "dianaloginkan"	Revisi	
5 Mei 2020	Bab 3: Revisi indentasi judul tabel. Buat rata kiri	Revisi	
8 Mei 2020	Acc bab 3 dan 4, lnjutkan bab 5 dan daftar pustaka	Disetujui	
1 Mei 2020	Perbaiki penggunaan jenis penomoran poin di bab 5. Seragamkan spt yg sudh dijelaskan di awal ttg hirarki penomoran sub judul dan poin.	Revisi	
1 Mei 2020	Lengkapi laporan mulai dr cover smpai akhir dlm satu file.	Revisi	
4 Mei 2020	Siapkan berkas utk semhas	Revisi	
27 Juni 2020	Lengkapi laporan, acc sidng	Disetujui	
7 Agustus 2020	Acc jilid	Disetujui	

Medan, 20 Agustus 2020
Dosen Pembimbing,

Ika Devi Perwitasari, S.Kom., M.Kom

YAYASAN PROF. DR. H. KADIRUN YAHYA
UNIVERSITAS PEMBANGUNAN PANCA BUDI
LABORATORIUM KOMPUTER
Jl. Jend. Gatot Subroto Km 4,5 Sei Sikambing Telp. 061-8455571
Medan - 20122

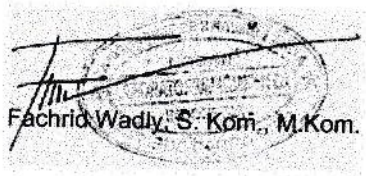
KARTU BEBAS PRAKTIKUM

ng bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

ma	: prendi parluhutan.s
P.M.	: 1614370320
gkat/Semester	: Akhir
ultas	: SAINS & TEKNOLOGI
usan/Prodi	: Sistem Komputer

ar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 08 Mei 2020
Ka. Laboratorium



Fachrid Wadly, S. Kom., M.Kom.

YAYASAN PROF. DR. H. KADIRUN YAHYA
PERPUSTAKAAN UNIVERSITAS PEMBANGUNAN PANCA BUDI
Jl. Jend. Gatot Subroto KM. 4,5 Medan Sunggal, Kota Medan Kode Pos 20122

SURAT BEBAS PUSTAKA
NOMOR: 1904/PERP/BP/2020

Kepala Perpustakaan Universitas Pembangunan Panca Budi menerangkan bahwa berdasarkan data pengguna perpustakaan atas nama saudara/i:

Nama	: prendi parluhutan.s
N.P.M.	: 1614370320
Tingkat/Semester	: Akhir
Fakultas	: SAINS & TEKNOLOGI
Jurusan/Prodi	: Sistem Komputer

Bahwasannya terhitung sejak tanggal 05 Mei 2020, dinyatakan tidak memiliki tanggungan dan atau pinjaman buku sekaligus tidak lagi terdaftar sebagai anggota Perpustakaan Universitas Pembangunan Panca Budi Medan.

Medan, 05 Mei 2020
Diketahui oleh,
Kepala Perpustakaan,

Sugiarjo, S.Sos., S.Pd.I

ABSTRAK

PRENDI PARLUHUTAN.S

Analisis Serangan SQL Injection Pada Web Server Menggunakan Intrusion Detection Systen

2020

Analisa *SQL Injection* adalah tindakan yang harus dilakukan *administrator network* dalam mengetahui sumber serangan yang terjadi pada *server.* Tindakan *preventif* perlu dilakukan bila diketahui banyaknya serangan yang *high priority* dari berbagai ancaman pada suatu *server*. Pada penelitian dilakukan skenario penyerangan pada suatu *server* yang dirancang, serangan yang dilakukan berupa serangan dengan teknik *SQL injection*. Teknik *SQL injection* yang diimplementasikan menggunakan teknik *Union-Based SQL Injection*. Pada Server juga telah disetting snort sebagai *Intrution Detection System* bila terjadi serangan maka fungsi *snort* akan bekerja dalam mendeteksi serangan *SQL Injection*. Dalam implementasi penelitian ini serangan *SQL injection* dengan metode memanfaatkan *operator SQL UNION* untuk mengkombinasikan hasil dari dua atau lebih perintah *SELECT* ke dalam satu hasil yang kemudian dikembalikan sebagai bagian dari *HTTP response*. dan menjadikan *attacker login* kedalam *database* tanpa akses *administrator*, akan tetapi hal ini terdeteksi oleh *system snort* yang aktif.

Kata Kunci: *SQL Injection, Intrusion Detection System (IDS), Web Server.*

ABSTRAK

PRENDI PARLUHUTAN.S

Analisis Serangan SQL Injection Pada Web Server Menggunakan Intrusion Detection Systen

2020

SQL Injection analysis is an action that must be done by network administrators in knowing the source of attacks that occur on the server. Preventive action needs to be taken if there are many high priority attacks from various threats on a server. In the study carried out an attack scenario on a server that was designed, the attacks carried out in the form of attacks with SQL injection techniques. SQL injection techniques are implemented using Union-Based SQL Injection techniques. The server has also set a snort as an Intrution Detection System when an attack occurs, the snort function will work in detecting SQL Injection attacks. In the implementation of this research SQL injection attacks using the method of using the SQL UNION operator to combine the results of two or more SELECT commands into one result which is then returned as part of the HTTP response. and makes the attacker log into the database without administrator access, but this is detected by the active system snort.

Kata Kunci: *SQL Injection, Intrusion Detection System (IDS), Web Server.*

DAFTAR GAMBAR

Halaman

DAFTAR ISI

BAB III METODE PENELITIAN

BAB IV HASIL DAN PEMBAHASAN

DAFTAR TABEL

Halaman

KATA PENGANTAR

Puji syukur kepada Tuhan yang Maha Esa karena dengan berkat dan kasih anugerah-Nya penulis masih diberikan kesehatan sehingga akhirnya penulis dapat menyelesaikan Skirpsi dengan judul : **"ANALISA SERANGAN SQL INJECTION PADA WEB SERVER MENGUNAKAN INTRUCTION DETECTION SYSTEM".**

Dalam penyusunan Skirpsi ini penulis menyadari banyak mengalami kesulitan namun berkat bantuan dan dorongan dari berbagai pihak, akhirnya Skirpsi ini dapat juga diselesaikan. Penulis dengan segala kerendahan hati menyampaikan terima kasih kepada:

1. Ayahanda dan Ibunda beserta keluarga yang telah berjasa dalam memberikan dukungan moril dan materil.
2. Bapak H.M. Isa Indrawan, SE, MM, selaku Rektor Universitas Pembangunanan Panca Budi Medan.
3. Rektor I, Bapak Ir. Bhakti Alamsyah, M.T, Ph.D
4. Bapak Hamdani ST., MT, selaku Dekan Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan
5. Bapak Eko Hariyanto, S.Kom.,M.Kom, selaku Ketua Program Studi Sistem Komputer Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan.
6. Dosen Pembimbing 1, Bapak Dian Kurnia S.Kom.,M.Kom
7. Dosen Pembimbing 2, Ibu Ika Devi Perwitasari S.Kom.,M.kom
8. Seluruh Dosen dan Staf Pegawai Fakultas Sains Dan Teknologi yang telah banyak membantu dalam kelancaran seluruh aktivitas perkuliahan.
9. Teman-teman yang telah memberikan berbagai saran, inspirasi, dorongan, doa, motivasi dan moril maupan materil yang diperlukan sehingga penulis dapat menyelesaikan Skirpsi ini.

Penulis juga menyadari bahwa penyusunan Skripsi ini belum sempurna baik dalam penulisan maupun isi disebabkan keterbatasan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk penyempurnaan isi Skirpsi ini.

Medan, Mei 2020
Penulis,

PRENDI PARLUHUTAN.S
NPM : 1614370320

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seseorang dalam melakukan serangan menggunakan *Injeksi SQL (Structured Query Language), hacker* akan memanfaatkan celah keamanan pada *website* atau *aplikasi*. melalui celah tersebut, hacker akan memasukkan perintah *SQL* berbahaya ke dalam *database* mesin *server* sehingga mereka dapat masuk ke dalam sistem tanpa *username* dan *password*, serangan *syber* ini dapat merusak *database* perusahaan anda. peretas dapat menggunakan *injeksi SQL* untuk menemukan *kredensial* pengguna lain di dalam *database* dan menyamar sebagai pengguna tersebut, serangan ini juga memungkinkan peretas untuk bisa mengubah dan menghapus *database* yang tersimpan dalam *sistem* anda dan penyerang dapat memanfaatkan *sintaks* dan kemampuan dari *SQL*, serta kekuatan dan *fleksibilitas* untuk mendukung fungsi *operasi database* dan fungsi *onalitas sistem* yang tersedia ke *database*. *Injeksi SQL* bukan merupakan kerentanan yang *eksklusif* mempengaruhi aplikasi *Web*, kode yang menerima masukan dari sumber yang tidak di percaya dan kemudian menggunakan input yang membentuk *SQL dinamis bisarentan (Clarke, 2009)*. Kasus *SQL Injection* terjadi ketika seorang penyerang dapat memasukkan serangkaian pernyataan *SQL ke query* dengan memanipulasi data input keaplikasi *(Anley,2002)*.

Berdasarkan defenisi tersebut, dapat dikatakan bahwa serangan *SQL Injection* sangat berbahaya karena penyerang yang telah berhasil memasuki

database system dapat melakukan manipulasi data yang ada pada *database* sistem. Proses manipulasi data yang tidak semestinya oleh penyerang dapat menimbulkan kerugian bagi pemilik *website* yang terinjeksi. kebocoran data dan informasi merupakan hal yang fatal. Data-data tersebut dapat disalah gunakan oleh pihak yang tidak bertanggung jawab. keamanan data dan informasi sangat penting dalam menjaga ketahanan sebuah *website*. Berdasarkan uraian-uraian tersebut, maka dinilai perlu untuk menguji kemanan *website* kita terhadap serangan *SQL Injection,* serta melaku kan analisa terhadap kelemahan sistem yang ada, sehingga dapat diperoleh tindakan selanjutnya untuk perbaikan sistem.

Penulis mengemukakan bahwa *SQL injection* adalah jenis aksi *hacking* pada keamanan komputer di manaseorang penyerang bias mendapatkan akses ke *basis data* di dalam sistem. *SQL Injection* juga yaitu serangan yang mirip dengan serangan *XSS* dalam bahwa penyerang memanfaatkan aplikasi *vector* dan juga dengan *Common* dalam serangan *XSS*. *SQL injection* juaga merupakan salah satu kelemahan yang paling dahsyat untuk dampak bisnis, karena dapat menyebabkan pembongkaran semua informasi yang sensitif yang tersimpan dalam sebuah aplikasi *database*, termasuk informasi berguna seperti *username,password,* nama, alamat, nomor telepon, dan rincian kartu kredit.

Web server adalah sebuah perangkat lunak atau *software* yang memberikan pelayanan berbasis data yang digunakan untuk menerima permintaan atau *request* dari pengguna internet atau biasa disebut *client* berupa *http* atau *https* yang kemudian akan ditampilkan ke dalam bentuk halaman *website*. *Web server* selain

sebagai perangkat lunak atau *software* juga dapat digolongkan dalam perangkat keras atau *hardware*.

Web server sebagai *software* berfungsi melayani permintaan *client* sedangkan sebagai *hardware*, *web server* berfungsi sebagai tempat penyimpanan semua data. *Web server* dan fungsinya inilah yang membantu kita saat melakukan pencarian di internet, *Web server* dapat disebut untuk melakukan atau akan mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. halaman *web* yang diminta terdiri dari berkas teks, video, gambar, file dan lain-lainnya (Muttaqin et al., 2014).

Intrusion Detection System (IDS) adalah sebuah sistem yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah system atau jaringan. *Intrusion Detection System (IDS)* juga berfungsi sebagai pendeteksi masuk dan keluarnya kegiatan-kegiatan yang mencurigakan berhubungan dengan *traffic* jaringan. Selainitu *Intrusion Detection System (IDS)* berfungsi sebagai memberikan peringatan kepada *system* atau *administrator* (Nugroho et al., 2015).

Pada penelitian ini, peneliti memanfaatkan serangan *SQL injection* untuk melihat sejauh mana pengukuran dari keamanan data dari suatu simulasi *web server* yang kita buat pada *OS Ubuntu server Dan Kali Linux*. Peneliti mencoba melakukan serangan *SQL injection* terhadap *server* yang akan dibangun dan akan dilakukan dengan menggunakan beberapa *host* agar dapat memaksimalkan pengujian serangan terhadap *web server yang kita bangun pada Ubuntu server* ini, dan mengetahui ketahanan server dari serangan *SQL injection*.

Maka dari itu saya tertarik mengajukan skripsi dengan judul:

"Analisis Serangan SQL Injection Pada Web Server Menggunakan Intrusion Detection Systen".

1.2 Rumusan Masalah

Setelah menguraikan latar belakang di atas maka dapat di simpulkan masalah yang akan di selesaikan yaitu sebagai berikut:

1. Bagaimana *server* dapat mengidentifikasi serangan *SQL injection*?
2. Bagaimana merancang *instrusion detection system* pada *server* berbasis *ubuntu*?
3. Bagaimana pengamanan *web server* dari *serangan SQL injection*?

1.3 Batasan Masalah

Agar penelitian mendapatkan hasil yang di inginkan sesuai dengan rencana, Berikut adalah batasan masalah yang akan dibahas dalam penelitian ini, yaitu:

1. Dalam melakukan penyerangan ke *web server*, *web server* hanya menggunakan *software snort* untuk sebagai *intruction detection system(IDS)*
2. Contoh serangan yang di ambil yaitu *Union-Based SQL Injection.*
3. Komputer *attacker* menggunakan *system operasi parrot linux*
4. *Intrussion detection system* dibangun pada *system operasi linux ubuntu*
5. *Snort* di utamakan hanya *mengidentifikasi* serangan *SQL injection* dan *terdeteksi pada console terminal*

1.4 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Tujuan penelitian ini memberikan contoh serangan *SQL injection* pada *web server* yang harus kita waspadai.
2. Mengurangi ancaman serangan *SQL injection* pada *web server* dengan *intrussion detection system.*
3. Mengetahui bagaimana kinerja *web server* saat keadaan *web server* normal dan saat adanya serangan.

1.5 Manfaat Penelitian

Berikut adalah manfaat penelitian:

1. Dapat mengetahui seperti apa serangan *SQL injection* dan ketika serangan *SQL injection* menyerang *web server* seperti apa terjadi kendala yang di alaminya.
2. Mengetahui kinerja server yang sedang berjalan dan setiba di serang seperti apa aja kondisi *web server* tersebut.

BAB II

LANDASAN TEORI

2.1 Pengertian *SQL injection*

SQL injection adalah kerentanan yang terjadi ketika penyerang memiliki kemampuan untuk mempengaruhi *Structured Query Language (SQL) query* yang melewati suatu aplikasi ke *database back-end*. Dengan mampu mempengaruhi apa yang akan diteruskan ke *database*, penyerang dapat memanfaatkan *sintaks* dan kemampuan dari *SQL*, serta kekuatan dan *fleksibilitas* untuk mendukung fungsi operasi *database* dan fungsionalitas sistem yang tersedia ke *database*. *Injeksi SQL* bukan merupakan kerentanan yang *eksklusif* mempengaruhi aplikasi *Web*, kode yang menerima masukan dari sumber yang tidak dipercaya dan kemudian menggunakan *input* yang membentuk *SQL* dinamis bisa rentan (Clarke, 2009). Kasus *SQL Injection* terjadi ketika seorang penyerang dapat memasukkan serangkaian pernyataan *SQL* ke *query* dengan memanipulasi data input ke aplikasi (*Anley, 2002*).

Berdasarkan defenisi tersebut, dapat dikatakan bahwa serangan *SQL Injection* sangat berbahaya karena penyerang yang telah berhasil memasuki *database sistem* dapat melakukan manipulasi data yang ada pada *database sistem*. proses manipulasi data yang tidak semestinya oleh penyerang dapat menimbulkan kerugian bagi pemilik *website* yang *terinjeksi*. Kebocoran data dan informasi merupakan hal yang fatal. data-data tersebut dapat disalahgunakan oleh pihak yang tidak bertanggung jawab. keamanan data dan informasi sangat penting dalam

menjaga ketahanan sebuah *website.* berdasarkan uraian-uraian tersebut, maka dinilai perlu untuk menguji kemanan *website* terhadap serangan *SQL Injection*, serta melakukan analisa terhadap kelemahan sistem yang ada, sehingga dapat diperoleh tindakan selanjutnya untuk perbaikan system (Universitas et al., 2013).

2.2 Pengertian Client Server

Client Server merupakan model *konektivitas jaringan* yang berfungsi menjadikan komputer menjadi *client* dan *server*, pada *konektivitas* ini sebuah komputer ditempatkan sebagai *server* dan sebuah komputer lainnya menjadi *client* dimana *server* berfungsi untuk mengelola data dan melayani jalur atau *terminal* yang terhubung pada sebuah sistem jaringan yang dapat dikatakan sebagai *client*. dan sebaliknya sebuah *client* tidak dapat menjadi *server*, akan tetapi *server* dapat menjadi *client* yang dimana prinsip kerja dari *server* akan menunggu permintaan *client*, memproses dan membagikan hasilnya kepada *client* sedangkan client mengirimkan permintaan kepada *server* untuk diproses dan mendapatkan hasil prosesnya (Nanang et al., 2008).

2.3 Pengertian Jaringan

(Pamungkas, 2016) menjelaskan jaringan komputer merupakan sekelompok komputer otonom yang saling berhubungan antara satu dengan lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi. Saat ini jaringan komputer bukan merupakan hal yang baru. Setiap *instansi*, telah memanfaatkan jaringan komputer. Penggunaan

jaringan komputer menjadi sangat meningkat dikarenakan kebutuhan akan informasi yang menjadi semakin tinggi.

(Varianto, Mohammad Badrul, 2015) menjelaskan jika dilihat berdasarkan luas area yang dapat dijangkau atau dilayani jaringan komputer terbagi menjadi 3 jenis yaitu *LAN, MAN*, dan *WAN*. *Klasifikasi* Jaringan Komputer adalah sebagai berikut:

2.4 Jaringan LAN

(Varianto, Mohammad Badrul, 2015) menjelaskan *LAN* adalah jaringan komputer yang jaringannya hanya mencakup wilayah kecil, seperti jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil. Saat ini, kebanyakaan *LAN* berbasis pada teknologi *IEEE 802.3 Ethernet* menggunakan perangkat switch, yang mempunyai kecepatan *transfer* data 10, 100, atau 1000 Mbit/s. selain teknologi *Ethernet*, saat ini teknologi 802.11b (atau biasa disebut *Wifi*) juga sering digunakan untuk membentuk *LAN* dengan teknologi *Wifi* biasa disebut *hotspot*.

2.5 Jaringan MAN

(Varianto, Mohammad Badrul, 2015) menjelaskan *MAN* adalah sebuah jaringan komputer besar yang mencakup sebuah kota atau sebuah kampus besar. *MAN* biasanya merupakan gabungan dari *LAN* yang menggunakan teknologi *backbone* berkecepatan tinggi dan menyediakan layanan ke jaringan yang lebih besar seperti *WAN* dan *internet*. *Metropolitan Area Network* (*MAN*) adalah

jaringan dalam suatu kota dengan *transfer* data berkecepatan tinggi, yang menghubungkan berbagai lokasi seperti kampus, perkantoran, pemerintahan, dan sebagainya. Jaringan *MAN* adalah gabungan dari beberapa *LAN*. Jangkauan *MAN* ini antara 10 hingga 50 km, *MAN* ini merupakan jaringan yang tepat untuk membangun jaringan antara kantor-kantor dalam suatu kota antara pabrik/instansi dan kantor pusat yang berada dalam jangkauannya, prinsip sama dengan *LAN*, hanya saja jarak lebih luas yaitu 10-50 km.

2.6 Jaringan WAN

(Varianto, Mohammad Badrul, 2015) menjelaskan suatu *WAN* meliputi area *geografi* yang lebih luas lagi, yang meliput suatu negara atau dunia. Umumnyan jaringan ditempatkan pada lokasi yang berbeda. *WAN* digunakan untuk menghubungkan banyak *LAN* yang secara *geografis* terpisah. *WAN* dibuat dengan cara menghubungkan *LAN* menggunakan layanan seperti *Leased Line, dial-up,* satelit atau layanaan paket *carrier*. Dengan *WAN*, sekolah yang ada di Indonesia dapat berkomunikasi dengan sekolah yang ada di *Munchen Jerman* dalam beberapa menit saja tanpa mengeluarkan biaya yang banyak. *Wide Area Network (WAN)* merupakan jaringan komputer yang mencakup area yang besar sebagai contoh yaitu jaringan komputer antar wilayah, kota, atau bahkan negara, atau dapat didefinisikan juga sebagai jaringan komputer yang membutuhkan *router* dan saluran komunikasi publik.

2.7 Web Server

Web server dikenal dapat melayani permintaan pengguna berupa *http* dari *client* yang terhubung dalam jaringan dan memberikan pelayanan kepada yang meminta informasi berkaitan dengan *website* dan memberikan suatu hasil berupa halaman *web* yang ditampilkan dalam *browser. Web server* menggunakan *port* 80, *web server* sendiri terdiri dari dua komponen, yang pertama adalah komputer dan *software web* server yang digunakan, dimana pada *web server* inilah *website* yang digunakan untuk memberikan informasi atau bertukar informasi ditempatkan.

Aplikasi *web server* dapat diperoleh dengan mudah baik yang berbayar maupun tanpa bayar. Saat akan memilih perangkat lunak aplikasi *web server*, *administrator web* harus memilih *web server* manakah yang akan digunakan untuk melayani para pengguna website institusinya, *administrator* melakukan ini ketika akan melakukan *hosting server* untuk digunakan *website* institusi. Berkaitan dengan pelayanan *server,* membutuhkan aplikasi pada sistem computer (*computer server dedicated*) yang berfungsi melayani permintaan akses dari komputer pengguna.

Beberapa aplikasi pelayanan *server* antara lain *Web Server, FTP Server, DHCP Server, Mail Server, DNS Server, FTP Server,* dan *Database Server.* Bila *web server* dan *website*-nya yang berisi tampilan informasi-informasi dapat diakses menggunakan *web browser* seperti *Mozilla Firefox* atau *Google Chrome.* Berikut merupakan jenis *web server* antara lain: *Apache Web Server, Apache Tomcat, MS Windows server 2003* Internet Information *Service (IIS), Light HTTP, Sun Java System Web Server, Zerus Web Server* serta *Nginx*. Studi implementasi

hosting server, khusus fokus pada bagian implementasi *web server* akan bertemu pada masalah bagaimana kinerja *web server* itu sendiri. Dan studi kinerja dari *web server* pada permasalahan menganalisis kinerja *web server* yang akan diimplementasi. Studi *web server* akan bertemu dengan berbagai aplikasi perangkat lunak *web server* seperti tersebut di atas. Analisis mengangkat pembandingan *web server Apache* dengan *web server Nginx*. Beberapa hal dasar yang perlu diketahui berkaitan dengan pembandingan aplikasi *web server* yang dianalisis. *Hosting Server, Apache HTTP server, web server Nginx,* dan hasil studi pembandingan yang telah dilakukan beberapa peneliti lainnya, studi pembandingan kinerja *web server* menjadi menarik. Kontribusi analisis *web server* untuk pengembangan *hosting server* institusi, khusus fokus pada bagian implementasi *web server* yang bertemu pada masalah bagaimana kinerja *web server*, berujung pada pembandingan kinerja *web server*:

1) Memberikan alternatif pemilihan *web server* yang dapat melayani kecepatan transfer data, waktu *request*, dan koneksi,

2) Memberikan solusi bagi *administrator* untuk menentukan topologi jaringan.

Dan terakhir, gambaran dan penyesuaian penggunaan *web server* dilihat dari kelebihan dan kekurangannya akan diperoleh *administrator*. *Hosting server* merupakan komputer khusus yang terhubung dengan internet secara *real time*, dan secara terus-menerus agar pengguna internet dapat mengakses perangkat lunak yang menjadi *backbone* dari *World Wide Web* dikenal dengan *web server* merupakan perangkat lunak *server*. *Web server*, merupakan perangkat lunak untuk

berkomunikasi dengan *client (web browser)* dan mempunyai protokol sendiri yaitu *Hyper Text Transfer Protocol*. Dengan protokol ini, komunikasi antar *web server* dengan *client-nya (browser)* dapat saling dimengerti dan lebih mudah. Proses yang dimulai dari permintaan *client (browser),* diterima *web server*, diproses, dan dikembalikan hasil prosesnya oleh *web server* ke *web client* lagi dilakukan secara transparan. *HTTPS* memiliki pengertian yang sama dengan *HTTP* hanya saja *HTTPS* memiliki fungsi di bidang keamanan *(secure). HTTPS* menggunakan *Secure Socket* Layer *(SSL)* atau Transport Layer *Security (TLS)* sebagai sub layer dibawah *HTTP* aplikasi layer. *HTTP* di-enkripsi dan deskripsi dari halaman yang diminta oleh pengguna dan halaman yang dikembalikan oleh *web server*. Kedua protocol tersebut memberikan perlindungan yang memadai dari serangan *eaves droppers* dan *man in the middle attacks.* Pada umumnya *port* yang digunakan *HTTPS* adalah *port* 443. Tingkat keamanan tergantung pada ketepatan dan mengimplementasikan pada *browser* dan perangkat lunak *server* dan didukung oleh algoritma penyandian yang aktual. Oleh karena itu, pada halaman *web* digunakan *HTTPS* dan *URL* yang digunakan dimulai dengan *https://.*

Apache HTTP server adalah perangkat lunak dengan *platform oprating system (OS)* yang mendukung *multi-tasking*, dan menyediakan layanan untuk aplikasi lain yang terhubung ke dalamnya, seperti *web browser*. *Apache* pertama kali dikembangkan untuk bekerja dengan *sistem operasi Linux/Unix*, tetapi kemudian diadaptasi untuk bekerja di bawah sistem lain, termasuk *Windows* dan *Mac*. *Nginx* adalah *software* open-*source* yang memiliki kinerja tinggi sebagai

server HTTP dan *reverse proxy*. *Nginx* dengan cepat memberikan konten *statis* dengan penggunaan efisien sumber daya *sistem*. Hal ini dapat menyebarkan dinamis *HTTP* konten di jaringan menggunakan *FastCGI handler* untuk *script*, dan dapat berfungsi sebagai perangkat lunak yang sangat mampu penyeimbang beban. *Nginx* dibangun secara *modular* dan dengan demikian mampu mendukung berbagai fitur seperti *Load Balancing* dan *Reverse Proxying*, *Virtual hosts* berbasis nama dan *IP*, *Fast CGI*, akses langsung ke *cache*, *SSL*, *Flash Video Streaming* dan sejumlah fitur-fitur standar lainnya. *Nginx* dapat dijalankan dan tersedia untuk *platform Unix*, *Linux*, *varian dari BSD*, *MacOS X*, *Solaris*, dan *Microsoft Windows*

2.8 Pengertian *Snort*

Snort adalah perangkat lunak yang gratis dan *open source* dalam melakukan *intrusion detection* dan *prevention system* yang dibuat oleh *Martin Roesch di 1998*. *Snort* memiliki kemampuan untuk melakukan lalu lintas *real-time* analisis dan pencatatan paket pada *Internet Protocol* (IP) jaringan. Ini melakukan analisis protokol, pencarian konten, dan pencocokan konten. program ini juga dapat digunakan untuk mendeteksi *probe* atau serangan, tetapi tidak terbatas pada, upaya *operating system fingerprinting attempts*, *common gateway interface*, *buffer overflows*, *server message block probes*, dan *stealth port scans* (Sistem & Dan Manajemen, 2018).

Snort adalah paket *sniffer* berbasis *libpcap* dan *logger* yang dapat digunakan sebagai *Intrusion Detection System* (IDS). Aturan ini berdasarkan *logging* untuk

melakukan pencocokan pola konten dan mendeteksi berbagai serangan, seperti *buffer overflow*s, *stealth port scans*, *CGI attack*, *SMB probes*, dan banyak lagi. *Snort* memiliki peringatan secara *real-time* yang mampu memberitahukan peringatan yang dikirim ke syslog, *Server Message Block* (SMB) pesan *WinPopup,* atau *file alert* yang terpisah (Dewi, 2017).

Dalam penggunaan *Snort* masih di basiskan *command-lin*e dapat memberikan cukup kesulitan bagi pengguna yang terbiasa akan lingkungan *Graphical User Interface* (GUI) sehingga ada beberapa pihak pendukung yang mengembangkan seperti *Acid, Mircrosoft Windows*, dan *IDS center* dengan *basis php* yang dapat di akses menggunakan *web browser* berbasis *GUI.*

Terdapat 4 mode dalam menjalankan Snort sebagai analisa keamanan jaringan yaitu :

1. Mode *Logger*

   Jenis mode ini bekerja dalam mencatat semua data yang dilalui pada jaringan untuk dapat di analisa di lain hari.

2. Mode *Sniffer*

   Jenis mode yang bekerja dengan menangkap dan melihat data yang melewati pada jaringan

3. Mode *Intrusion Detection*

   Jenis ini membuat Snort bekerja sebagai pendeteksi kegiatan terhadap berbagai serangan pada jaringan yang mengikuti berdasarkan aturan *(Rules).*

4. Mode *Inline*

   Jenis yang bekerja aktif sebagai pengaman pada jaringan dengan memblokir upaya penyerangan dan memberikan respon terhadap serangan menggunakan *Rules* yang di kombinasikan dengan *Firewall* untuk di izinkan atau tidak data pada jaringan berdasarkan *Rules* yang telah di tentukan.

2.9 Rules Snort

Rules Snort merupakan *database* yang berisi pola-pola serangan yang berupa *signature* jenis serangan, *Rules* Snort ini harus secara rutin di*update*. sehingga ketika terjadi pola serangan baru, *Snort* dapat mendeteksi pola tersebut sebagai sebuah serangan. Penulisan *rules* Snort mempunyai aturan yaitu *rules* harus ditulis dalam satu baris (*single line*) (Khamphakdee et al., 2015).

Untuk membaca dan membuat *rules* pahami bagian yang ada pada *rules* terdiri dari *Rules Header* dan *Rules Options*. *Rules Header* berisikan tindakan, protokol, alamat IP, nomor port, *destinasi host*. *Rules Options* berisikan pesan yang akan di tampilkan nantinya, dapat di jelaskan dari 1 perintah *Rules* seperti berikut :

```
Alert tcp any any -> $HOME_NET (msg:”ICMP DETECT”;)
```

Dari *Rules* di atas dapat dilihat untuk *Rules Header* di tandai dari Alert tcp any any -> $HOME_NET dan untuk *RulesOptions* isi pesan yang ada pada di antara (msg:”ICMP DETECT”;).

2.10 Data AcQuisition (DAQ)

Data AcQuisition (DAQ) adalah modul tambahan untuk paket *Input / Output* pada *Snort* yang digunakan untuk mengaktifkan fitur *Pervention* yang ada pada *Snort*.

Berikut beberapa *modul DAQ* untuk mendukung *Snort* agar dapat berjalan dengan fitur *Prevention* :

1. Packet *CAP ture* (PCAP) dapat menjadikan *Snort* bekerja secara default dengan *Sniffer* dan *Intrusion Detection System* (IDS).
2. *AFPACKET* menjadikan Snort bekerja pada mode *inline* dengan menggunakan 2 *interface* yang saling menjembatani tanpa ada tambahan *Firewall.*
3. *NetFilter Queue (NFQ)* menjadikan *Snort* bekerja secara *inline* dengan menggunakan *Queue dan netfilter atau Firewall.*
4. *IPFW* membuat *Snort* bekerja dengan mode *inline* untuk *Open BSD* dan *FreeBSD* menggunakan *socket pf* dan *ipfw.*
5. *DUMP* memungkinkan *Snort* melakukan pengujian mekanisme *inline* dan normalisasi.

2.11 Intrusion Detection System (IDS)

Intrusion Detection System (IDS) adalah perangkat lunak yang digunakan untuk memonitor jaringan untuk apa pun aktivitas yang tidak menyenangkan menjembatani fungsi normal sistem sehingga menyebabkan beberapa pelanggaran kebijakan. ini meninjau beberapa *sistem deteksi intrusi* dan perangkat lunak yang

menyoroti klasifikasi utama mereka, evaluasi kinerja dan pengukuran (Nugroho et al., 2015).

Intrusion detection system dapat di kalsifikasi menjadi 3 bagaian yaitu :

1. *Host Intrusion Detection System* (HIDS)

   Jenis ini ditempatkan pada satu perangkat seperti *server* atau *workstation,* dimana data dianalisis secara lokal ke mesin dan mengumpulkan data ini dari berbagai sumber.*HIDS* dapat menggunakan sistem deteksi *anomali* dan penyalah gunaan.

2. *Network Intrusion Detection System* (NIDS)

   NIDS dikerahkan pada titik strategis dalam jaringan infrastruktur.*NIDS* dapat menangkap dan menganalisis data mendeteksi serangan yang diketahui dengan membandingkan pola atau tanda dari *database* atau deteksi aktivitas ilegal dengan memindai lalu lintas untuk aktivitas anomali. *NIDS* juga disebut sebagai "*Packet-sniffer*", karena ini menangkap paket yang lewat melalui media komunikasi.

3. *Hybrid Intrusion Detection System*

   Manajemen dan memperingatkan dari kedua perangkat deteksi intrusi jaringan dan berbasis *host,* dan menyediakan pelengkap logis untuk *NID* dan *HID* - manajemen *deteksi intrusi* pusat (Ashoor dan Gore, 2011).

Dalam kebanyakan *IDS* adalah sistem yang bersifat pasif yang mana tugas dari *IDS* ini hanyalah mendeteksi *intrusi* yang bila terjadinya penyerangan dan memberikan peringatan kepada *admin* jaringan bahwa terjadinya penyerangan.

2.12 Jenis – Jenis Serangan SQL injection

Serangan *SQL injection* memiliki beberapa jenis penyerangan dengan beberapa cara yaitu :

a. *Classical SQL Injection* adalah cara yang paling sering dilakukan dengan menggunakan metode *UNION* dalam menggabungkan dua *query* untuk menampilkan informasi penting dari *database*. sebelum melakukan injeksi menggunakan *UNION*, seorang *attacker* sudah harus memahami bagaimana *query* yang akan dieksekusi untuk mendapatkan informasi yang penting di dalamnya. seorang *attacker* dapat memasukkan *malicious charater* seperti *single quote tag, double minus* dan sebagainya untuk menghasilkan pesan *error*, dimana pesan *error* tersebut akan dimanfaatkan untuk mengeksploitasi *web* tersebut.

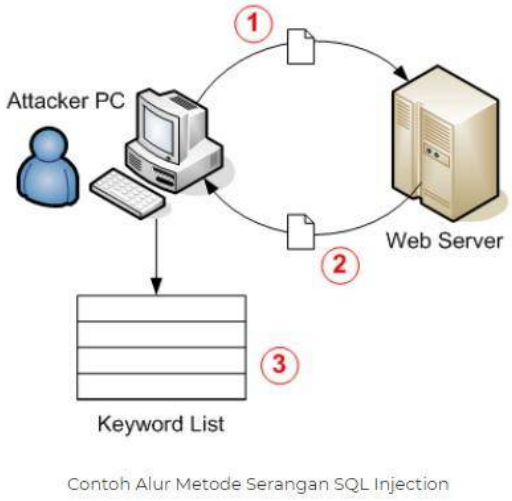
b. *Blind SQL Injection sering disebut juga dengan true/false yaitu ketika injeksi dilakukan berhasil tetapi tidak menampilkan pesan error kepada attacker, melainkan kembali ke halaman itu sendiri atau menampilkan sebagian, seluruh konten maupun tidak menampilkan konten apapun dari web. teknik ini membutuhkan waktu yang lama dikarenakan menebak informasi yang terdapat di dalam database dan respon dari request berupa true/false. jika url yang dimanipulasi mengasilkan nilai true, maka akan menampikan konten tetapi jika url yang dimanipulasi menghasilkan nilai false, maka tidak akan memproses request dari attacker.*

c. *Double Blind SQL Injection / Time-based ketika injeksi terhadap url gagal menggunakan teknik blind sql injection, tidak menutup kemungkinan web tersebut memiliki celah terhadap sql injection. Injeksi kemungkinan sukses dilakukan*

tetapi dapat ditangani di dalam database sehingga ketika injeksi yang dilakukan berhasil, tetapi hasilnya tidak dapat dilihat di dalam aplikasi dan tidak terlihat oleh attacker. Teknik double blind sql injection yaitu berupa gabungan antar blind sql injection/classical sql injection dengan penundaan waktu. jika url yang digabungkan dengan time delay akan menghasilkan perintah sesuai dengan request yang diminta, maka aplikasi web tesebut dapat di injeksi menggunakan serangan SQL injection.

Analisis penanganan *SQL Injection* dilakukan dengan cara pada saat inisialisasi sebuah *variable* di kode pemograman ataupun pada kode program yang mengambil data dari *database (query),* dilakukan *validasi* terhadap karakter – karakter berbahaya seperti *single quote*, *double quote*, *comment*, *tautology*, dan lain sebagainya membatasi panjang inputan yang akan dimasukkan sehingga *attacker* tidak dapat menginjeksi dengan memasukkan inputan yang panjang ke dalam *form login* Mengatasi pesan *error* yang keluar dari *database* dengan menghilang kannya atau menyembunyi kan pada kode program (arrissetiawan, 2013).

Contoh cara kerja serangan *SQL injection* sebagai berikut :



Gambar 2.1 Proses Penyerangan *SQL injection*
Sumber (*Otanaha*, 2018)

2.13 Paket Filtering Iptables

Iptables mengizinkan *user* untuk mengontrol sepenuhnya jaringan melalui paket IP dengan *system Linux* yang diimplementasikan pada *kernel Linux*. Sebuah kebijakan atau *Policy* dapat dibuat dengan *iptables* sebagai polisi lalulintas jaringan (Sondakh *et al*, 2014),.

Paket *filtering Iptables* adalah *tools* perangkat lunak yang biasanya di pergunakan oleh *linux* sebagai *filter traffic* atau lalu lintas data di dalam *server*. *sistem* yang dirancang khusus untuk mencegah akses serangan yang mencurigakan masuk ke dalam jaringan.

Paket *filter Iptables* memiliki aturan dasar, yaitu:

1. *Input*, semua paket data yang masuk dari *firewall* dari *intranet* atau *internet*.

2. *Output, semua paket data yang keluar dari firewall melalui intranet atau internet*
3. *Forward,* Paket data yang melalui *firewall* dari *intranet* atau *internet.*
4. *Prerouting,* digunakan sebagai mentranslasikan paket sebelum proses *routing* terjadi, yaitu merubah alamat tujuan dari paket data yang biasanya disebut dengan *Destination NAT* atau *DNAT.*
5. *Postrouting*

Digunakan sebagai mentraslasikan alamat setelah proses *routing* terjadi, yaitu merubah alamat dari paket data biasanya disebut dengan *Source NAT* atau *SNAT* (Pi, 2019).

2.14 MySQL

MySQL merupakan *DBMS (Database Management System)* yang di pergunakan secara gratis berlisensi dari *General Public License(GPL),* dimana setiap orang bebas untuk menggunakannya akan tetapi tidak untuk dijadikan program untuk komersial.

MySQL merupakan generasi lanjut dari *SQL (Structured Query Language). SQL* adalah sebuah konsep pengoperasian basis data terutama dalam proses seleksi, pemasukan, pengubahan, dan penghapusan data yang mungkin dapat di kerjakan dengan mudah dan otomatis (Kholid, n.d.).

Beberapa fitur yang terdapat pada *MySQL* yaitu:

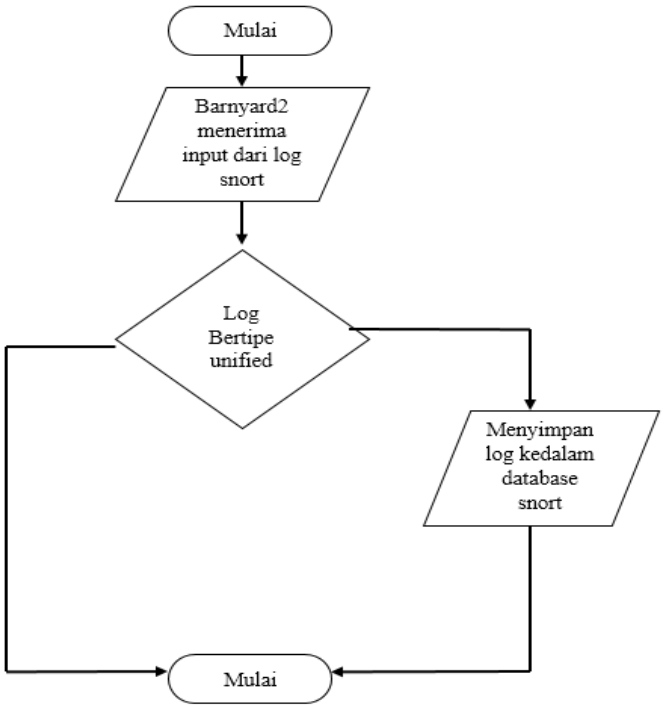
1. *RDBMS (Relational Database Management System).*
2. *Arsitektur dengan Client-Server.*

3. Perintah *SQL standar* mudah di kenal. hampir segala software database menggunakan *SQL*.
4. Mendukung *foreign key*.
5. Bebas dalam penggunaan atau Gratis.
6. *Fleksibel* dengan berbagai bahasa pemrograman.
7. Stabil dan keamanan yang baik.

2.15 Barnyard2

Barnyard adalah sebagai penghasil *output system* dari *Snort*. *Barnyard2* bekerja memproses kemudian menyimpan proses *output biner* dari *Snort* kedalam *database MySql*. *Barnyard2* akan membaca *file logging Snort* dan memasukkannya ke dalam *database* (S, 2015).

Untuk proses penulisan log penyerangan ke dalam *database* menggunakan *barnyard2* dapat dilihat dengan *flowchart* berikut:



Gambar 2.2 Proses Penulisan Log Masuk ke *Database*
Sumber (Hadi, 2016)

2.16 Linux

Linux adalah *sistem operasi open source* yang paling terkenal dan paling banyak digunakan. Sebagai *sistem operasi*,merupakan bagian dari *unix* yang bekerja pada berbagai macam perangkat keras komputer mulai dari *inter* x86 sampai dengan *RISC*. Dengan berlisensikan *GNU (Gnu Not Unix)* dengan memperoleh program dan dilengkapi dengan kode sumbernya *(source code).*Tidak hanya itu, dapat pula hak untuk menyalin sebanyak yangdimau, atau dapat mengubah kode sumbernya hingga hal tersebut legal dilakukan dibawah *lisensi.Lisensi GNU* memperbolehkan pihak yang ingin mendapatkan biaya untuk penyalinan maupun pengiriman program.Yang paling penitng pada *Linus* yaitu kebebasan, terutama untuk programmer dan *administrator* jaringan, yaitu

kebebasan untuk memperoleh kode sumber (source code) dan kebebasan untuk mengubahnya (Wamiliana et al., 2013).

2.17 Sistem Operasi Ubuntu

Ubuntu adalah berkumpulnya dari individu-individu yang umum dalam menciptakan sebuah *sistem operasi* yang bebas. *Sistem operasi open source* itu dapat disebut juga dengan *Ubuntu*. Sebuah *sistem operasi* dengan beberapa program dasar dan *utilitas* yang membuat komputer dapat berjalan. Inti dari sebuah *sistem operasi* tersebut dapat di sebut juga dengan kernel. Pengertian dari *kernel* adalah program yang paling dasar yang ada pada komputer dan memungkinkan kita untuk memulai program lain.

Kernel yang di gunakan *Ubuntu* saat ini menggunakan *kernel Linux* atau *kernel FreeBSD. Linux* yang dimulai dari *Linus Torvalds* dan didukung ribuan programmer di seluruh dunia. *FreeBSD* adalah sebuah *sistem operasi* termasuk *kernel* dan perangkat lunak lainnya.

Program perangkat lunak tersebut dapat membantu *brainware* dalam mendapatkan apa yang ingin mereka lakukan dan diperbuat, dari memodifikasi dokumen untuk menjalankan bisnis, bermain game dan menulis perangkat lunak yang lebih banyak lagi. *Ubuntu* datang dengan lebih dari 45.000 paket (*software precompiled* yang terbungkus dalam format yang bagus untuk memudahkan dalam *installasi* dan konfigurasi), manajer paket *(APT)* dan beberapa *utilitas* yang mungkin dapat untuk mengelola ribuan paket yang ada pada ribuan komputer dengan mudah untuk menginstal sebuah aplikasi. *Ubuntu* sangat mengatur segala

sesuatu pekerjaan di *sistem operasi*nya sehingga semua bekerja dengan sangat baik (Ngatmono et al., 2015).

2.18 Sistem Operasi Kali Linux

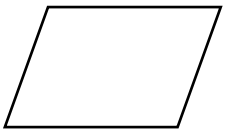
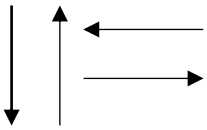
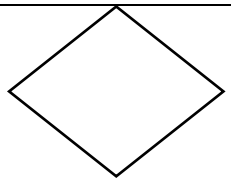
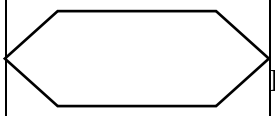
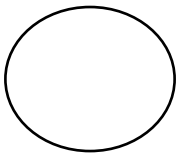
Kali Linux adalah distribusi berlandasan distribusi *Debian GNU/Linux* untuk tujuan *forensik* digital dan di gunakan untuk pengujian penetrasi, yang dipelihara dan didanai oleh *Offensive Security*. *Kali Linux* secara teori bisa di katakan *Backtrack* versi 6. *Kali Linux* adalah salah satu distro *Linux* yang sangat terkenal dan juga diakui sejak kemunculannya pada tahun 2006 silam sebagai *sistem operasi* terbaik untuk keamanan *sistem* serta hacking. biasanya *sistem operasi Kali Linux* ini dijadikan sebagai *sistem operasi* paling utama bagi para dunia *hacker underground, sistem operasi* ini telah dilengkapi dengan 300 lebih senjata *hacking* yang sangat *kompatibel*, alias belum ditemukan penangkalnya (Bhatt, 2018).

2.19 Flowchart

Flowchart adalah bagian-bagian yang memiliki arus dan menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan suatu cara penyajian dari suatu Algoritma (Subrata, 2015)

Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses didalam program. Simbol-simbol yang digunakan dapat dibagi, yakni sebagai berikut:

Tabel 2.1 Simbol Flowchart

Simbol	Keterangan
	Input/Output Digunakan untuk mewakili data input/output
	Arus/Flow Digunkana untuk menunjukkan arah/alir dari suatu proses.
	Proses Digunakan untuk mewakili suatu proses.
	Keputusan/*Decision* Digunakan untuk suatu penyelesaian kondisi dalam program.
	Persiapan/*pendefined* Proses Digunakan untuk memberikan nilai awal dari proses.
	Penghubung/*Connector* Digunakan untuk menunjukkan sambungan dari aliran yang terputus dihalaman yang sama.
	*Predefined* proses Digunakan untuk proses yang detilnya terpisah.
	Awal/akhir (Terminal) Digunakan untuk menunjukkan awal dan akhir dari proses.

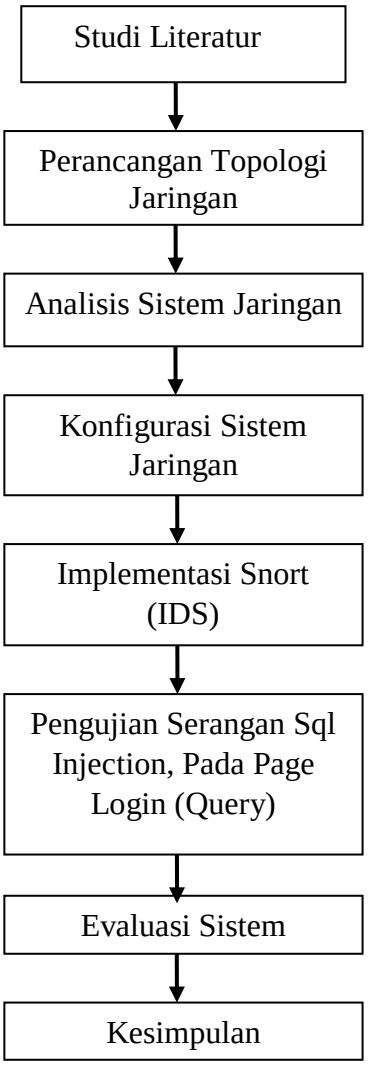
Sumber : Jogiyanto HM (2000 : 662)

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Metode yang digunakan dalam membangun sistem ini adalah model *waterfall*. Model ini terdiri dari beberapa tahapan, yaitu: *Studi Literatur*, perancangan *sistem*, konfigurasi *sistem*, penerapan *sistem*, *dokumentasi* dan *evaluasi sistem*. Adapun metode perancangan adalah sebagai berikut :



Gambar 3.1 Tahapan Penelitian Menggunakan Waterfall

Dalam menyelesaikan skripsi ini penulis memperoleh data dengan menggunakan beberapa tahapan-tahapan dari gambar 3.1 sebagai berikut:

1. *Studi Literatur*

   Dengan pengumpulan data-data berupa teori baik dengan dosen pembimbing maupun dengan orang yang berkompeten dalam kasus ini dan pustaka yang mendukung.

2. Perancangan *Sistem*

   a. Meliputi *beberapa tahap yang terstruktur sebagai berikut :*

   Sistem dirancang menggunakan *sistem operasi* dan konfigurasi *linux Ubuntu Server,* serta menggunakan paket-paket pendukung *linux Ubuntu Server,* seperti *bind9 apache2 php5-mysql mysql-server phpmyadmin libapache2-mod-php5* dan *snort,* menggunakan *sistem* operasi *windows* sebagai pengujian dan *parrot linux* sebagai penyerang.

   b. Hasil dan pembahasan dengan cara implementasi perangkat dan pengujian sistem.

3. *Konfigurasi Sistem*

   Dalam skripsi ini *sistem* yang dikonfigurasi yaitu menggunakan *linux ubuntu, bind9 apache2 php5-mysql mysql-server phpmyadmin libapache2-mod-php5* dan *snort sebagai pengidentifikasi* seranngan pada *web server* yang dapat mendeteksi adanya serangan dan sebagai himbauwan *masuknya serangan, untuk monitoring web server* menggunakan *console Terminal*.

4. Pengujian *Sistem*

   Melakukan pengujian dan penaksiran ulang *sistem* yang telah melakukan *implementasi.*

5. Dokumentasi dan Evaluasi System

   Apakah sistem yang telah di miliki mendapatkan kinerja yang baik dan keamanan dengan tingkat yang baik.

3.2 Metode Pengumpulan Data

Untuk mendukung penelitian yang akan dibangun dibutuhkannya metode pengumpulan data. Beberapa teori yang ada pada *website, jurnal, makalah*, dan penelitian lainnya. Dalam tahap ini juga melakukan *analisis sistem* yang berjalan dan kebutuhan *sistem* yang nantinya akan dikonfigurasi, dan *data* yang dikumpulkan haruslah data yang benar. Ada beberapa metode yang dapat digunakan dalam pengumpulan data, yang akan dibahas di bawah ini

1. Pengamatan (Observasi)

   Pengamatan atau *observasi* adalah sebuah metode pengumpulan data dengan melakukan pengamatan langsung kepada objek penelitian untuk melihat dari dekat kegiatan yang dilakukan. Dimana kegiantan ini dilakukan dengan mencatat informasi yang dilihat dan diketahui.

2. Wawancara

   Wawancara adalah bentuk komunikasi langsung antara peneliti dan responden. Wawancara juga merupakan suatu cara pengumpulan data yang digunakan untuk memproleh informasi langsung dari sumbernya,

wawancara iyalah berdasarkan sifat pertanyaan, maka dapat dibagi menjadi tiga, yaitu : wawancara terpimpin, wawancara bebas dan wawancara bebas terpimpin. Dan dari bentuk pertanyaannya dibagi menjadi tiga bentuk lagi, yaitu : wawancara berstruktur, wawancara tak berstruktur dan campuran.

3. Dokumentar

   Dokumentar adalah catatan tertulis tentang berbagagi kegiatan atau peristiwa pada waktu yang lalu. Semua dokumen yang berhubungan dengan penelitian yang bersangkutan perlu dicatat sebagai sumber informasi

3.3 Analisis Sistem Yang Sedang Berjalan

Seseorang dalam melakukan serangan *Injeksi SQL (Structured Query Language)* akan lebih dipermudah ketika *web server* tidak mempunyai keamanaan system, maka dari itu kita perlu membangun sebuah *system* keamanan identifikasi pada *web serve*r yang berupa *Snort (IDS)* agar *hacker/attacker* akan meninggalkan jejak masuk kedalam *system* jaringan *web server*.

SQL Injection adalah sebuah *metodologi* serangan yang menargetkan data yang berada dalam *database* melalui *firewall* yang melindungi *data* tersebut dan *SQL Injection* terjadi ketika seorang penyerang dapat memasukkan serangkaian pernyataan *SQL* ke *query* dengan *memanipulasi data input* ke *aplikasi* tersebut.

Intrusion Detection System (IDS) adalah sebuah metode yang dapat mengidentifikasi atau mendeteksi aktivitas yang mecurigakan pada *web server*.

Snort adalah salah satu *tools open source* yang digunakan yang semula *snort* bekerja sebagai *Intrusion Detection System* (IDS), dengan tambahan paket *filtering Iptables* dan didalam *Snort* terdapat modul tambahan *DAQ NetFilterQueue* (NFQ) sebagai *Prevention* bagi *Snort* yang nantinya berfungsi sebagai pendeteksi adanya *traffic* keluar masuk pada *web server*, dan akan terdeteksi oleh *Snort. Snort* menggunakan *Console Terminal* dalam mengelolah data-data kejadian atau *security* untuk mudah dibaca dan menampilkannya kedalam basis *web interface*. *Console Terminal* nanti akan memperlihat informasi pada *web server*, bila *attacker* melakukan *penetration testing* pada *web server* .

1. Cara Kerja *Snort Intrusion Detection System* (IDS)

   Cara kerja dari *Snort* (IDS) yaitu ketika paket masuk ditangkap oleh *libpcap* melewati paket *filtering Iptables* kemudian untuk memaksimalkan paket yang masuk paket tersebut akan dicocokkan satu per satu dengan *rule Iptables* tetapi paket yang masuk dapat disaring dulu dengan *rule Iptables*, kemudian paket yang lolos dialihkan ke *NetFilter Queue* untuk diolah *Snort Engine* bila paket terdeteksi adanya suatu serangan maka *console terminal Snort* akan di gunakan untuk merekam alamat *IP* penyerang yang telah terdeteksi..

   Gambaran system kerja *Snort IDS* (*Intrusion Detection System*) dapat digambarkan sebagai berikut:

   a. (libpcap) *Library Packet Capture*

      Lipcap bekerja dalam menangkap dan memisahkan paket data yang melalui *Ethernet card* yang selanjutnya akan digunakan *Snort*.

b. *Filtering* Iptables

Iptables bekerja dalam menyaring paket yang masuk dengan packet tersebut nantinya tidak dicocokkan satu per satu dengan *rules Iptables* akan tetapi langsung diteruskan ke *Netfilter Queue*.

c. *Netfilter Queue*

Bekerja dengan mengolah paket yang masuk untuk diolah oleh *Snort* nantinya.

d. *Packet Decoder*

Packet Decoder bekerja Dengan memisahkan *Data Link, Protocol* IP, paket TCP dan UDP *Snort* memiliki informasi *protokol* yang akan di proses lebih lanjut.

e. *Preprocessor*

Preprocessor adalah komponen yang bekerja dalam menyusun atau mengubah paket data sebelum menuju ke *detection engine* dan beroperasi untuk mencari tahu bila paket data terjadi serangan.

f. *Detection Engine*

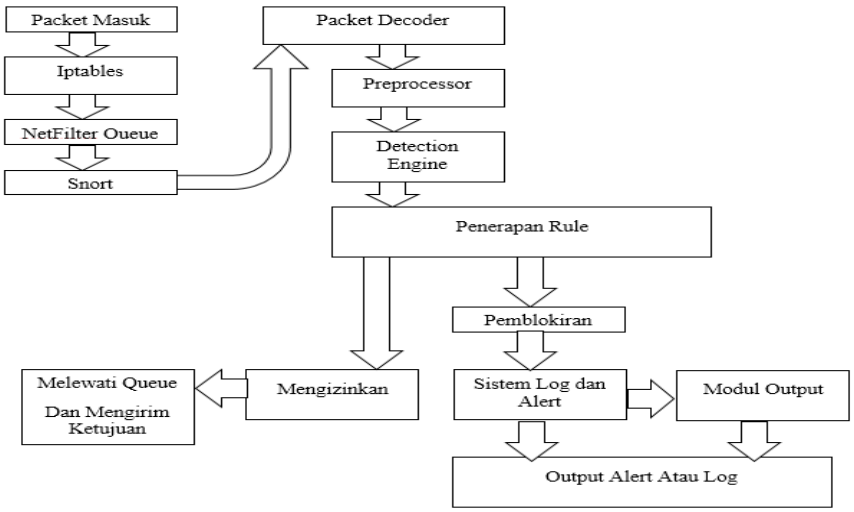
Detection Engine adalah bagian penting *snort*. Bekerja dengan mendeteksi bila terjadinya kegiatan penyerangan pada paket. *Detection Engine* memproses *rule Snort* untuk membaca *strutur data internal* yang di cocokkan dengan paket yang ada. Bila paket cocok dengan *rule* yang ada, tindakan yang di ambil berupa *logging* paket atau *console terminal* akan di biarkan saja.

g. Penerapan *Rules*

Telah di dapati oleh *Detection Engine* bila paket cocok dengan *rule* yang ada, tindakan yang di ambil berupa memisahkan paket apakah paket tersebut berupa ancaman atau tidak, bila itu ancaman *logging* paket pada *console terminal* dan log disimpan pada format teks didalam penyimpanan dan memblock ancaman tersebut dan bila tidak adanya tanda-tanda dari ancaman *logging* paket pada *console terminal* tidak terpicu berarti akses paket diizinkan.

h. Modul *Output*

Modul utama bekerja dengan mengatur jenis keluaran yang dihasilkan oleh sistem log dan *console terminal*



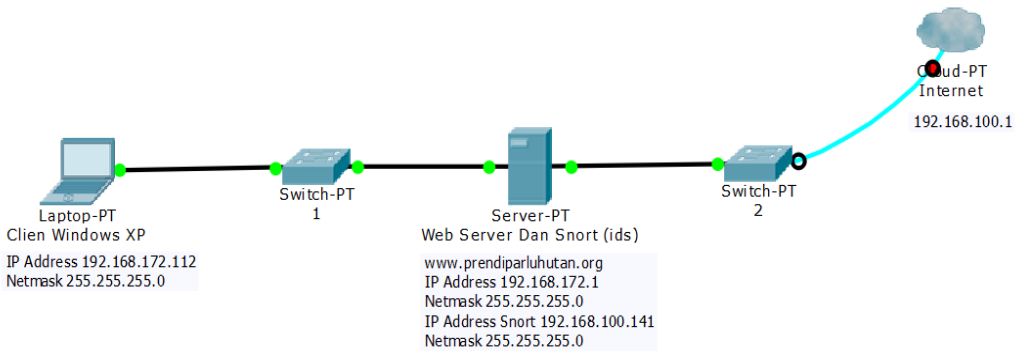
Gambar 3.2 Komponen Kerja Snort *IDS*

3.4 Rancangan Penelitian

3.4.1 Layout Jaringan Komputer

Dalam tugas akhir ini akan mengkonfigurasi sebuah serangan *SQL Injection, web server* dan *security system* yang memiliki kemampuan dalam *memonitoring* jaringan, *mendeteksi (detection*) pada aktifitas mencurigakan didalam jaringan, *Console Terminal* nantinya berfungsi sebagai *output biner Snort* yang diproses dan disimpan kedalam *database MySQL*.

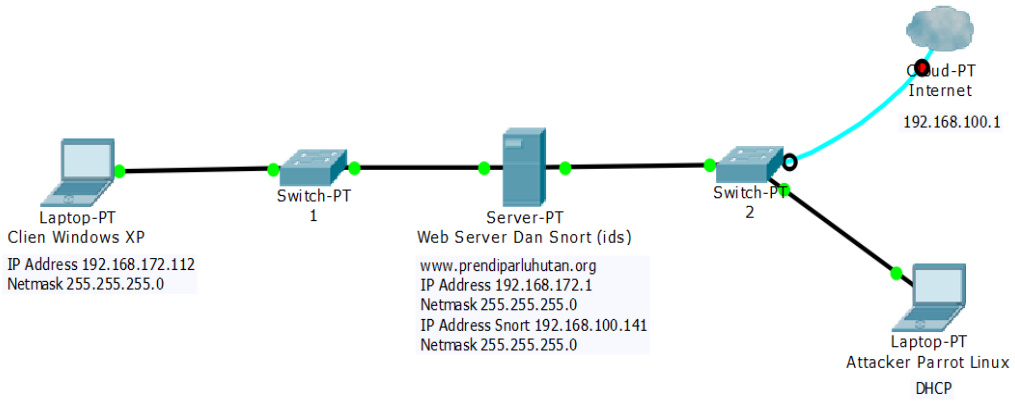
Sistem serangan *SQL injection* yang akan di bangun dapat digambarkan dengan topologi berikut:



Gambar 3.3 Topologi Sistem Sebelum Terjadi Serangan *SQL injection*

Adapun penjelasan dari topologi sistem jaringan diatas adalah sebagai berikut:

a. Pada *client* adalah suatu tempat pengujian hasil konfigurasi pada *web server*
b. *Switch* adalah suatu alat penghubung konektivitas pada jaringan menuju ke *web server* dan *internet*
c. Server adalah suatu alat untuk tempat konfigurasi *web server* dan *snort* (IDS)



Gambar 3.4 Topologi Sistem Sesudah Terjadi Serangan *SQL injection*

Adapun penjelasan dari topologi sistem jaringan diatas adalah sebagai berikut:

a. Pada *client* adalah suatu tempat pengujian hasil konfigurasi pada *web server*
b. *Switch* adalah suatu alat penghubung konektivitas pada jaringan menuju ke *web server* dan internet
c. Server adalah suatu alat untuk tempat konfigurasi web server dan snort (IDS)
d. *Attacker* berfungsi dengan tujuan untuk melakukan suatu serangan pada *server*

3.4.2 Anggaran Biaya

Untuk memenuhi dalam penelitian ini penulis melakukan pengumpulan biaya yang dikeluarkan untuk penelitian mengenai *Analisis Serangan SQL Injection* pada *Web Server* Menggunakan *Intrusion Detection Systen* sebagai berikut:

Tabel 3.2 Anggaran Biaya

No.	Hardware/Software	Spesifikasi	Jumlah	Harga
1.	Laptop untuk Client, Attacker dan Server	Intel Core i5 Ram 4Gb HDD 320 Gb	3	Rp. 5.300.000
2.	Cable UTP 1.5 Meter + 2 RJ45	Cat 5	1	Kabel Rp. 3000/Meter RJ45 Rp. 500
3.	Sistem Operasi : - Ubuntu Server - Parrot Linux - Windows 7	_	1 1 1	Rp. 15.000.00 Rp. 15.000.00 Rp. 15.000.00

3.4.3 Sistem Manajemen Jaringan

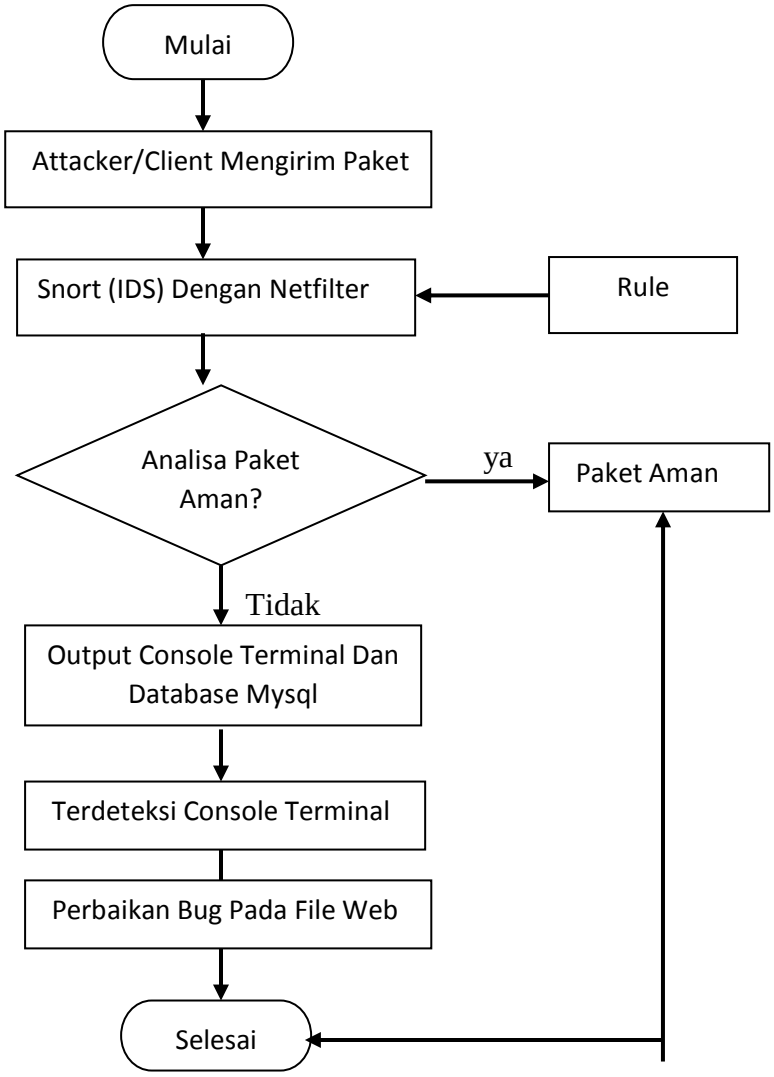
Dalam gambar 3.3 dan 3.4 diatas dapat dijelaskan dengan pengalamatan IP pada tabel berikut ini:

Tabel 3.1 Pengamatan Ip Address

NO	Hardware/Software Network	Port Ethernet	Alamat IP / IP Address
1	Sumber Internet	_	Address 192.168.43.1
2	Ubuntu Server	Eth0	Address 192.168.43.141 Netmask 255.255.255.0
	Web Server	Eth1	Address 192.168.172.1 Netmask 255.255.255.0
	Snort (IDS)	Eth0	Address 192.168.43.0/24
3	Attacker	Eth0	*Dynamic Host Configuration Protocol* (DHCP)
4	Client Terhubung Jaringan Lokal	Eth1	Address 192.168.172.25 Netmask 255.255.255.0 Gateway 192.168.172.1

Dalam tabel 3.1. Pengamatan alamat Ip dapat dejalaskan bahwa sumber internet berasal dari *hotspot* atau menggunakan *wifi* yang terhubung dengan *Server.* Kemudian didalam *ubuntu server* sudah terkonfigurasi sebuah *web server* dan konfigurasi *Snort IDS*, untuk pengamatan Ip pada *Server* dapat melakukan penyetingan jaringannya dengan menggunakan beberapa perintah.

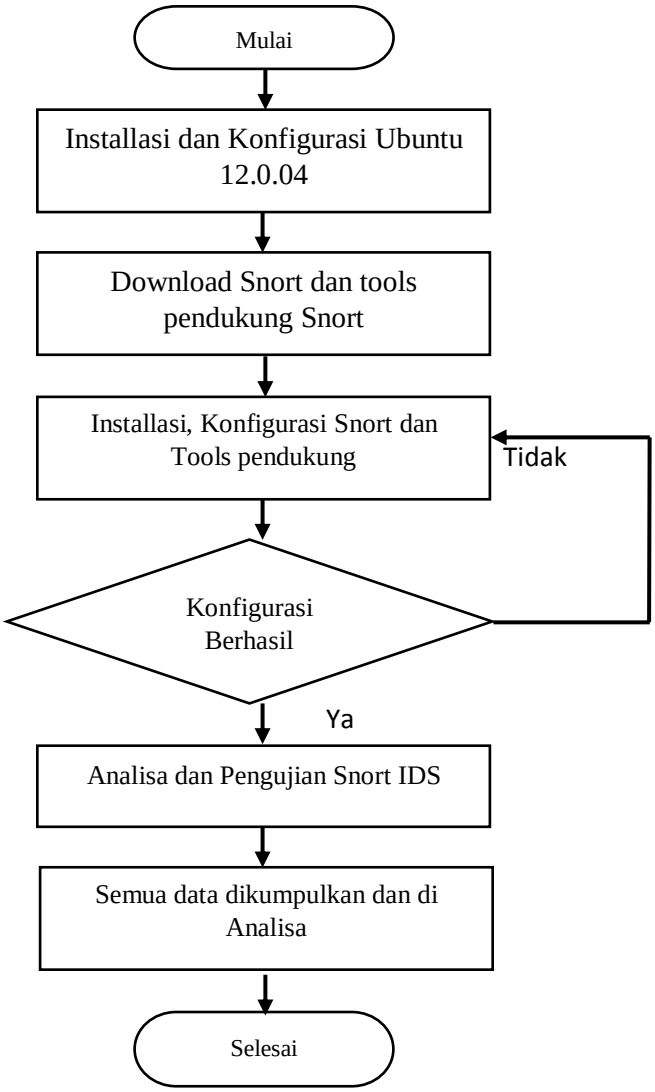
Flowchart Sistem serangan *Sql Injection, Snort (IDS)* dan *Web Server* dapat dilihat pada gamabar berikut:



Gambar 3.5 Flowchart Sistem *IDS* yang akan dibangun

3.4.4 Security Jaringan Snort (IDS)

Dalam membangun *snort* IDS (Intrusion Detection System) agar berjalan sesuai dengan apa yang di inginkan dengan baik, dibutuhkannya proses yang akan dibuat dalam bentuk diagram alir berikut :



Gambar 3.6 Flowchart Perancangan Konfigurasi *IDS*

Untuk penjelasan pada gambar diatas sebagai berikut :

1. Diawali dengan melakukan *installasi Linux Ubuntu 12.0.04* kemudian mengikuti alur *installasi* hingga selesai penginstallan. Saat telah selesai penginstallan lakukan penyesuaian *IP Address* dan konfigurasi pengroutingan.
2. Setelah selasai dalam penyetingan *IP Address* kemudian penginstallan paket-paket yang dibutuhkan dalam mendukung kinerja *Snort* agar penginstallan *Snort* nanti tidak terjadi kesalahan dengan menginstall paket-paket yang di butuhkan berupa *build-essential, libpcap-dev libpcre3-dev libdumbnet-dev, bison, flex, zlibb1g-dev, liblzma-dev, openssl, libssl-dev, autoconf, libtool. Pkg-config, mysql-server, libmysqlclient-dev, mysql-client, libcrypt-ssleay-perl, liblwp-useragent-determined-perl, apache2, libnetfilter-queue-dev, php5, dan tools php5* lainnya.
3. Bila semua tahap telah berhasil, lakukan tahap akhir yaitu pengujian sistem yang telah dibangun dan pengumpulan data dan menganalisa.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Spesifikasi Minimum Hardware dan Software

Specification Requirement adalah kebutuhan yang dibutuhkan dalam memenuhi kebutuhan spesifikasi pengaplikasian program aplikasi agar dapat berjalan dengan baik. *Specification requirment* terdiri dari dua bagian, yaitu kebutuhan perangkat keras (*hardware requirment*) dan kebutuhan perangkat lunak (*software requirment*).

1. *Hardware Requirement* dalam program aplikasi ini, penulis menggunakan laptop dan sistem operasi dengan spesifikasi sebagai berikut:
   1) Tipe Laptop : Dell Vostro 3400
   2) Processor : Intel(R) Core(TM) i5-3317U CPU @ 2.55GHz
   3) Memory : 4,00 GB (3,89 GB usable)
   4) Sistem Operasi : Windows 7 Pro 64 - bit
2. *Software Requirement* kebutuhan perangkat lunak (*software*) adalah:
   1) Virtual Box 6.0.14
   2) Winscp
3. *Sistem Operasi Requrement* kebutuhan pada penelitian adalah:
   1) Ubuntu Server
   2) Parrot Linux
   3) Windows 7/Windows Xp

4.2 Pengujian Aplikasi dan Pembahasan

Dalam hal ini sistem yang telah dianalisa dan dikonfigurasi dilanjutkan dengan sistem dioperasikan dan melakukan pengujian untuk melihat hingga sampai mana sistem yang dibuat dapat berjalan dengan baik hingga tujuan.

Dalam proses Implementasi Serangan *SQL Injection* menggunakan *Snort Instrusion Detection System* (IDS) terdapat bagian utama yang akan berperan yaitu:

1. Implementasi Serangan *SQL Injection*

   Bekerja untuk menyerang hak akses *database* agar dapat mendapatkan data dan informasi *website*

2. Implementasi *Web Server*

   Berperan sebagain objek tempat pengujian serangan *SQL Injection* dan notifikasi *Snort*

3. Implementasi *Rule Snort*

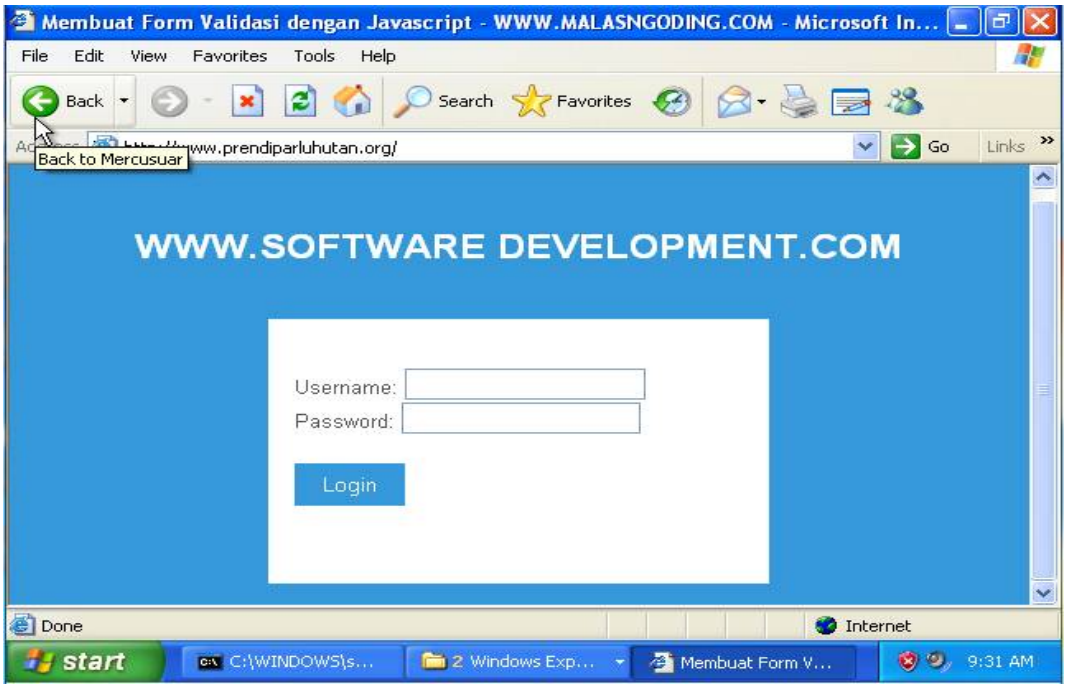
   Rule mendeteksi dan mengolah paket data yang melewati *Snort* apakah sebuah *Attacker* atau paket data tanpa ancaman

Implementasi Serangan *SQL Injection* menggunakan identifikasi *Snort Instrusion Detection System (IDS)* nantinya akan mendapatkan hasil dari identifikasi sebuah serangan yang terjadi pada server dan menampilkan *Output Web interface* pada *Console terminal*, Berupa paket *TCP* menggunakan perangkat lunak *open source SQLMAP*

1. Pembahasan Tampilan Website Target Secara umum

Tampilan dari *website* yaitu biasanya berupa *hypertext* (HTML) atau *hypermedia* yang dikirimkan ke users melalui *World Wide Web*. Untuk menampilkan suatu desain *web* atau isi dari suatu *website*, dibutuhkan sebuah *browser web* atau *software* (perangkat lunak) berbasis *web*. Tujuan dari *web* desain adalah untuk membuat *website* yang meliputi sekumpulan konten online termasuk dokumen dan aplikasi yang berada pada *web server*. Bisa juga, sebuah *website* berupa sekumpulan teks, gambar, suara dan konten lainnya, serta dapat bersifat interaktif maupun statis.

Pada tahap penelitian analisa ini *web server* hannya sebagi tempat pengujian serangan *SQL Injection website* http://www.prendiparluhutan.org



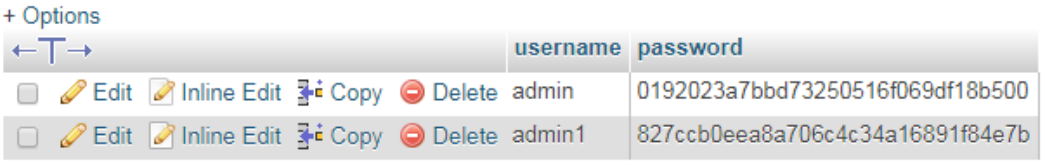
Gambar 4.1 Tampilan *Website* Target Secara Umum

2. Pembahasan Tampilan *Website* Target Beserta *Securitynya*

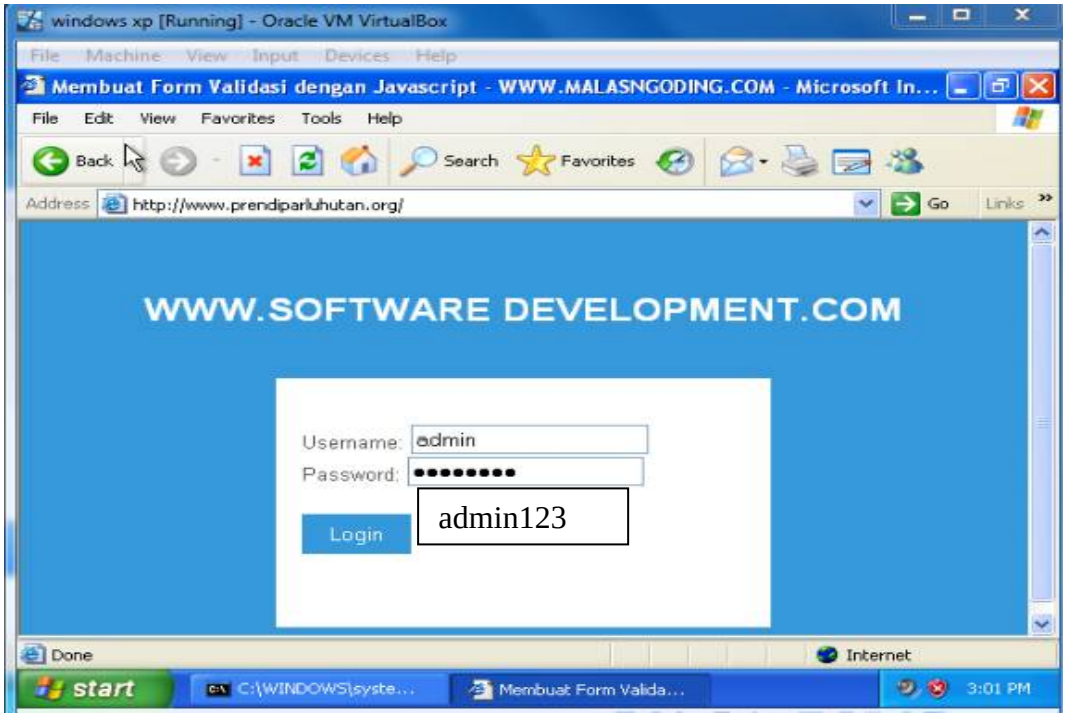
Tampilan pada *security website* target yang akan kita uji dan analisa yaitu dengan menggunakan *Security website* berbasis kriptografi *MD-5. MD-5* di dalam membangun keamanan aplikasi atau sistem informasi. Biasanya *MD5* di gunakan untuk meng-enkripsi data yang rahasia dan tidak ingin di minta oleh orang lain. contohnya yang paling sering adalah dalam membuat *login*. Hanya kata sandi data pada *basis data*, akan di enkripsi terlebih dahulu. agar jika pun ada orang yang tidak bertanggung jawab masuk dan dapat melihat isi *database*, maka tidak akan bisa menebak kata sandi yang diterima di sana. (karena sudah di enkripsi dengan MD5)

MD-5 memproses teks masukan ke dalam *blok- blok bit* sebanyak 512 bit, kemudian dibagi ke dalam 32 *bit sub blok* sebanyak 16 buah. Keluaran dari *MD-5* berupa 4 buah *blok* yang masing-masing 32 bit yang mana akan menjadi 128 bit yang biasa disebut nilai *hash.* Akibat pembagian ini, maka jumlah *biok* terakhir akan Iebih kecil atau sama dengan 512 bit. *Blok* terakhir tersebut akan mengalami *message padding.* Setelah proses *message padding,* jumlah bit pada *blok* terakhir adalah 448 bit.

Blok terakhir menghasilkan *output (message digest)* dari pesan tersebut. yaitu nilai dari buffer A, B, C dan D. Panjang *message digestnya* adalah 128 bit. Simpul utama MD5 mempunyai *blok* pesan dengan panjang 512 bit yang masuk ke dalam 4 buah ronde. Hasil keluaran dari MD-5 adalah berupa 128 bit dari *byte* terendah A dan tertinggi *byte* D.



Gambar 4.2 Tampilan *Security Login Website MD-5* Di Dalam *Database*



Gambar 4.3 Tampilan Penginputan *Sandi Login* Pada *Website*

3. Pengujian Serangan SQL Injection Dengan SQLMAP

Pengujian ini akan melakukan serangan menggunakan serangan SQL Injection pada perangkat lunak SQLMAP yaitu TCP (*Transmission Control* Protocol) yang bekerja dengan mengirimkan paket TCP dengan mencari kelemahan atau *exploitasi* pada sebuah *website* dan memaksa masuk ke dalam *databases server* sehingga dapat mampu mendapatkan informasi *databases* dan hak akses *databases*, serangan tersebut akan membuat si pemilik *web server* akan mendapatkan kerugian bersar karna informasi *databases* telah di ketahui *attacker.*

SQLMAP adalah alat uji *penetrasi open source* yang mengotomatisasi proses mendeteksi dan *mengeksploitasi* kelemahan *injeksi SQL* dan mengambil alih *basis data server*.

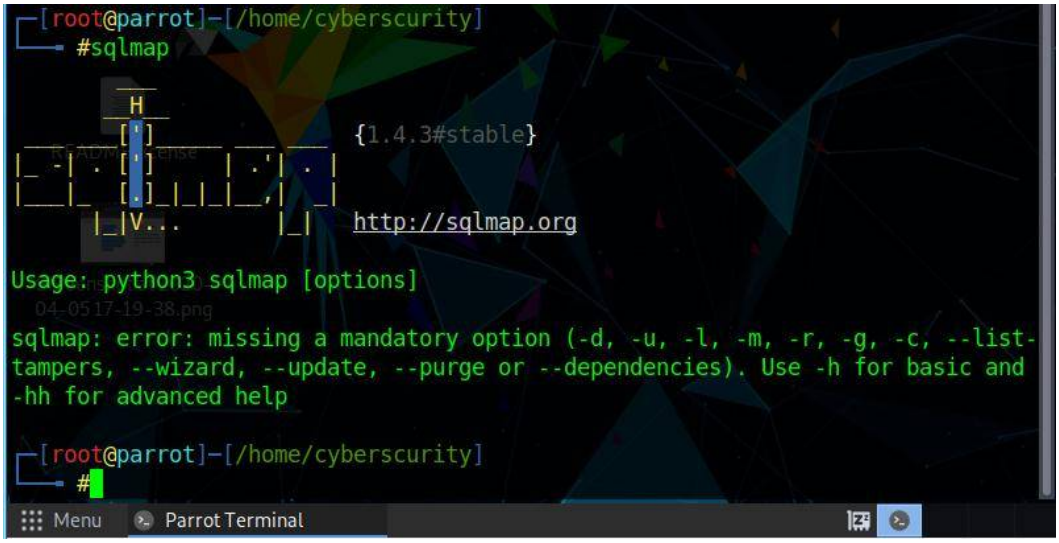
SQL injection sebuah teknik *hacking* di mana *attacker* dapat menyisipkan perintah-perintah *SQL* melalu *URL* untuk di eksekusi oleh *database*. *bug* atau *vulnerability* ini terjadi karena kelalaian seorang *programer* atau *webmaster* dalam melakukan pemograman *web* seperti tidak *difilternya variabel* dalam *web* tersebut.

dengan melakukan serangan *SQL injection* seorang *attacker* dapat mengambil alih serta memanipulasi sebuah *database* di dalam sebuah *server*, Berikut perintah pada *sqlmap* untuk melakukan serangan *SQL Injection.*

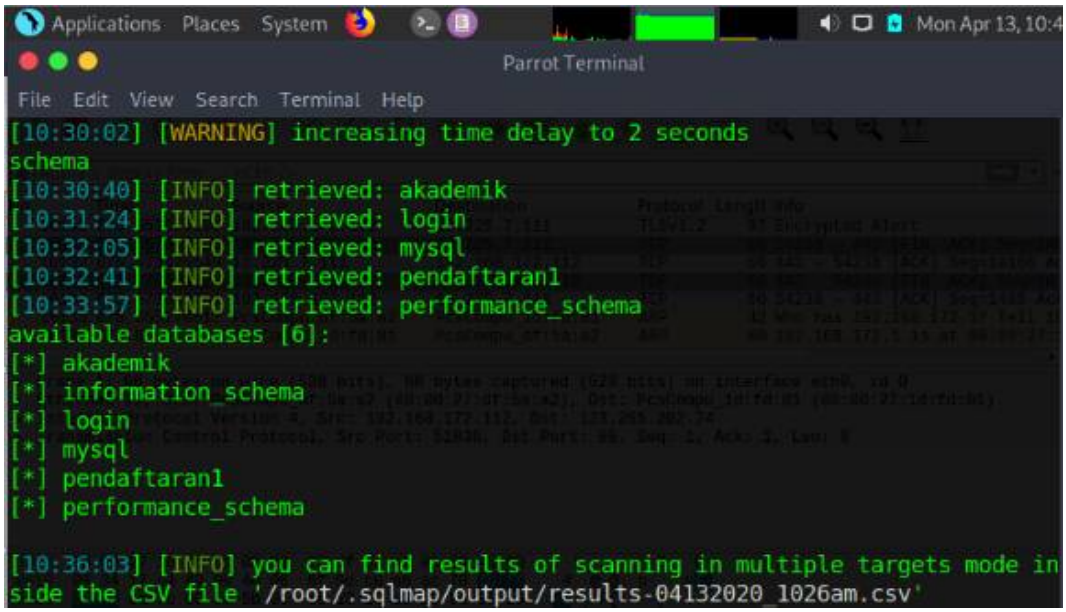
sqlmap -url http://www.prendiparluhutan.org --dbs --forms
sqlmap -url http://www.prendiparluhutan.org --dbs --forms -D akademik --tables

sqlmap -url http://www.prendiparluhutan.org --dbs --forms -D akademik -T admin --columns

sqlmap -url http://www.prendiparluhutan.org --dbs --forms -D akademik -T admin -C password,username --dump

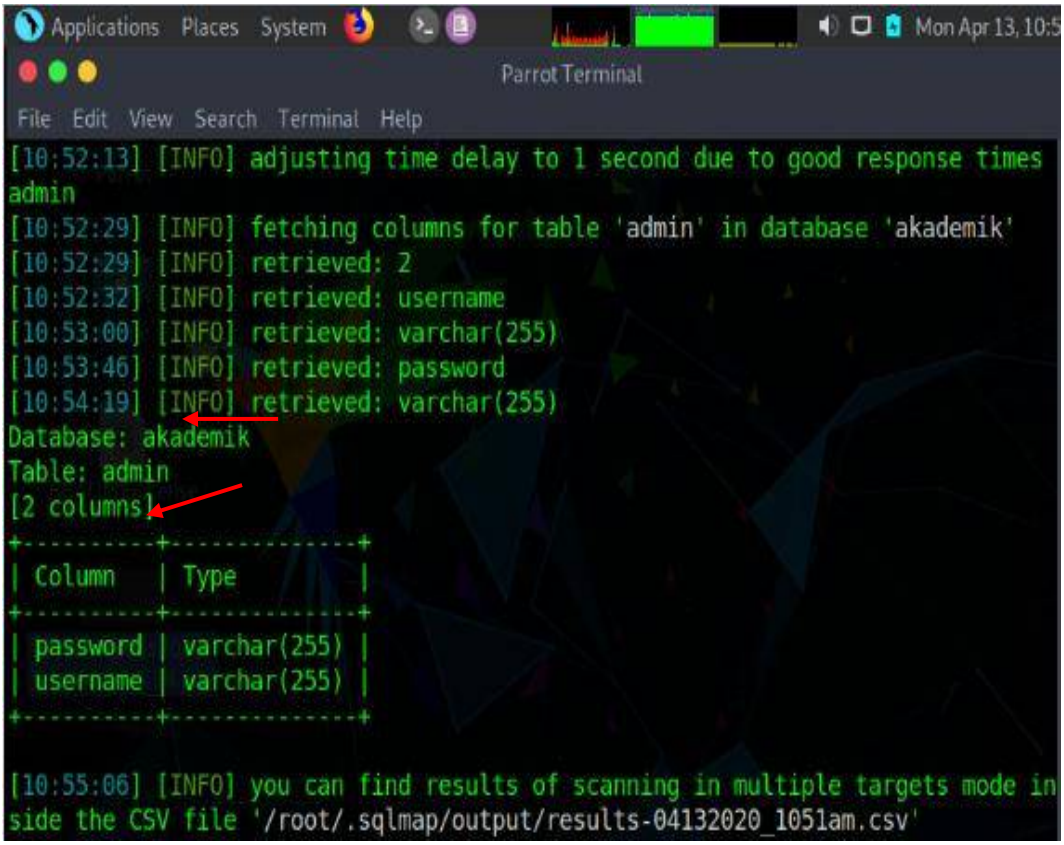


Gambar 4.4 Tampilan Utama SQLMAP



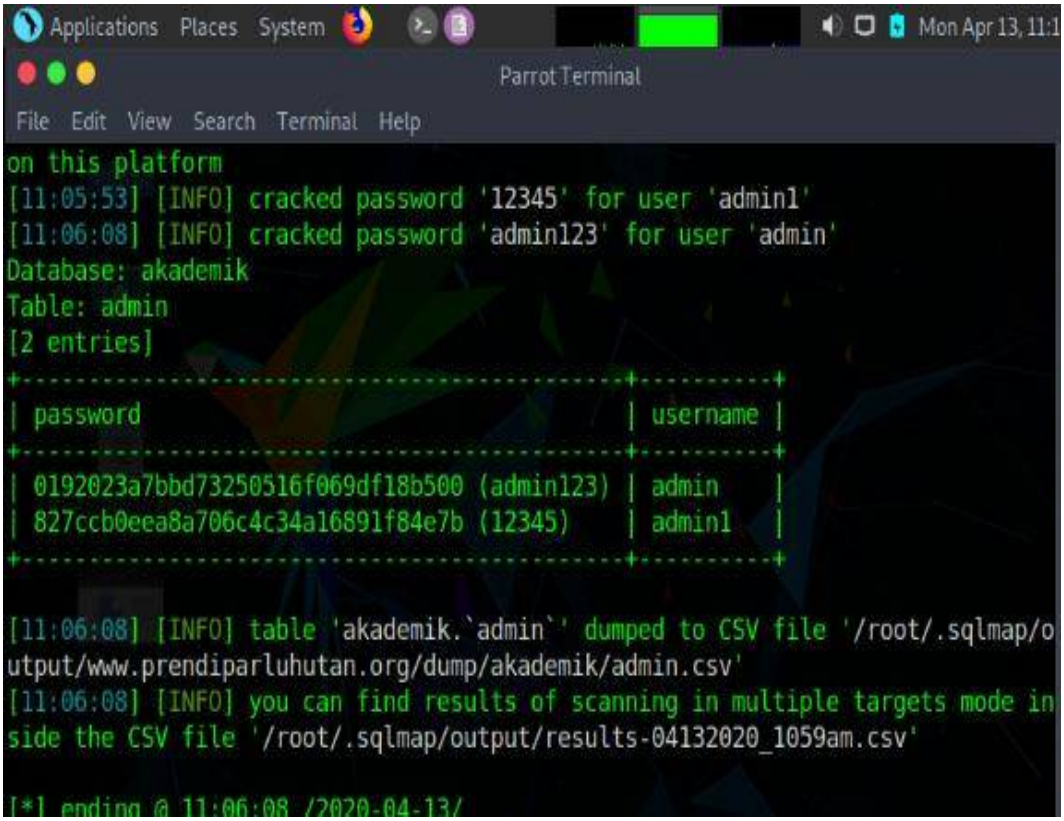
Gambar 4.5 Tampilan Informasi *Database*

Pada Gambar 4.5 kita sudah mendapatkan informasi nama *database* yang kita butuhkan melalu *exploitasi* atau kelemahan dari *website* menggunakan serengan *SQL Injection* pada *sqlmap*, dengan kelemahan pada *forms website* tersebut kita berhasil *mengexploitasikan akses administrator* pada *web server* sehingga kita dapat masuk kedalam *database* tanpa *security administrator web server* yaitu nama *database* akademik.



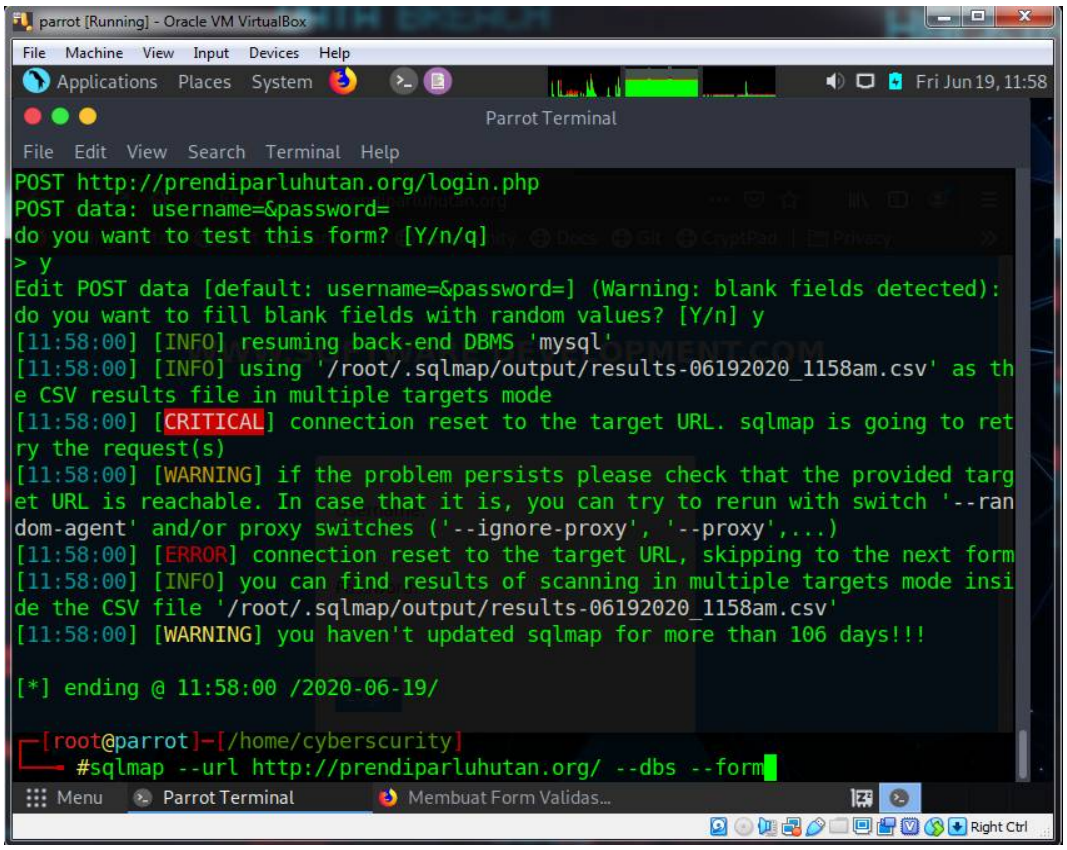
Gambar 4.6 Tampilan Informasi *Table* Dan *Columns Database*

Pada Gambar 4.6 ditahap ini kita suda berhasil memaksa masuk kedalam *database server website* dan kita sudah dapat mengetahui isi *database* yang berupa *table database* dan *colom database*.



Gambar 4.7 Tampilan Deskripsi *Password Md5 Database*

Pada tahap Gambar 4.7 disini kita sudah berhasil medapatkan isi *coloum* yang berupa *password* dan *username login* dengan menggunakan *methode securitty md-5* dan sudah dideskripsikan mengunakan *tools wordlist.txt* maka dari hasil deskripsi *password* dari *tools wordlist.txt* adalah *username* admin, *password* admin123 dan *username* 12345, *password* admin1



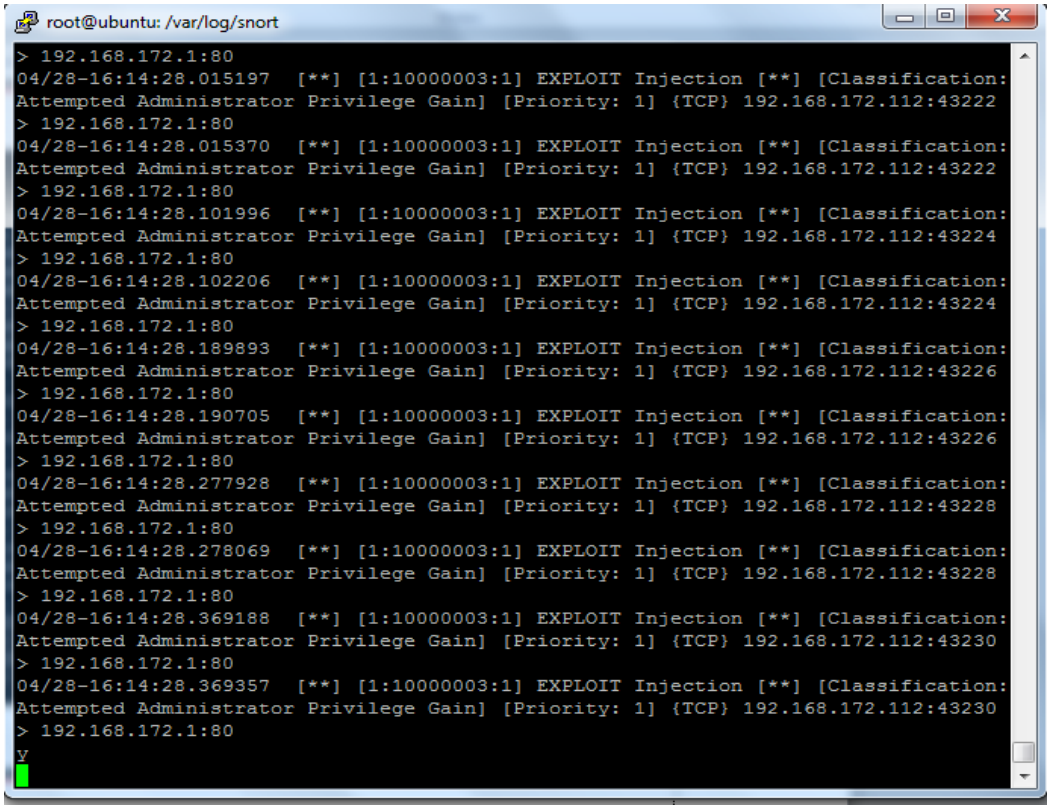
Gambar 4.8 Tampilan Hasil Pemblokiran Paket Serangan *SQL Injection* Menggunakan *IPS* Pada *Sqlmap*

Pada tahap gambar 4.8 disini *Sqlmap* dengan serangan *SQL Injection* tidak dapat melakukan aktifitas yang berupa serangan *SQL Injection* pada *server*, yang di sebabkan server telah *mengupgrade snort ids* menjadi *snort ips* dan *snort ips* telah melakukan pemblokiran paket serangan *SQL Injection* yang masuk pada *web server* dan mengakibatkan *attacker* keolahan dalam melakukan serangan ke *web server*, tetapi ini semuah hanyalah sebagai pencegahan yang di lakukan *web server*, *web server* juga tidak dapat menjaminkan keamanan yang dilakukan *snort ips* benar-benar aman pada *web server* dari serangan *SQL Injection* tingkat – tingkat tinggi lainnya.

4. Identifikasi Serangan Dengan Snort

Snort memiliki kemampuan untuk melakukan lalu lintas *real-time* analisis dan pencatatan paket pada *Internet Protocol* (IP) jaringan. Ini melakukan analisis protokol, pencarian konten, dan pencocokan konten. program ini juga dapat digunakan untuk mendeteksi *probe* atau serangan, tetapi tidak terbatas pada, upaya *operating system fingerprinting attempts*, *common gateway interface*, *buffer overflows*, *server message block probes*, dan *stealth port scans*.

Pada tahap ini *snort* akan berfungsi sebagai pendeteksi *traffic* keluar masuk yang berlebihan yaitu berupa suatu serang pada *web server* dan *snort* akan mengindentifikasi dengan menggunakan *alert* pada *console terminal* bahwasanya *web server* terjadi ada serangan.

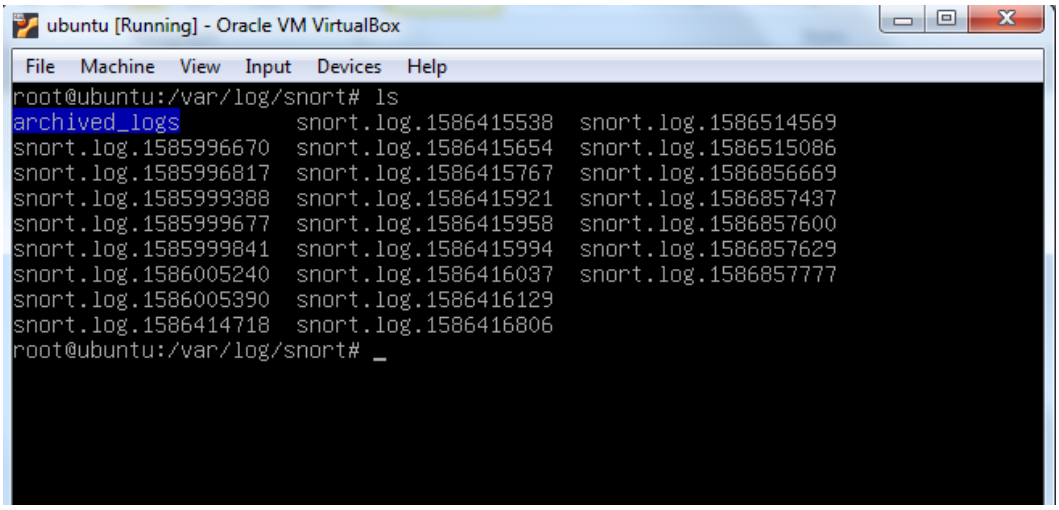


Gambar 4.8 Tampilan Identifikasi *Snort* Terjadinya Serangan

Pada Gambar 4.8 *snort* sudah mengidentifikasi bahwasanya *server website* sudah terjadi adanya serangan *SQL Injection* yang berupa paket TCP

5. Analisa Serangan Pada Log System Snort

Pengumpulan data *Network Forensic* menggunakan sebuah *tool* yang bisa menyimpan semua kejadian data-data lalu lintas jaringan. *Snort* merupakan *tool cross platform* yang mampu diinstall pada *Windows, Mac OS*, dan *Linux. Snort* merupakan *tool* yang *open source* dan *update* dari *Snort* dapat diakses oleh semua pengguna. *Snort* merupakan *tool* yang berbasis *Intrusion Detection System* (IDS) yang dapat memonitor jaringan yang berdampak serangan, selain itu juga menyimpan serangan tersebut pada Log. Tujuan dari penelitian ini yaitu mengimplementasikan *snort* serta menganalisa *Log snort* dari data serangan yang tersimpan pada *Log snort.*



Gambar 4.9 Tampilan Penyimpanan Data Identifikasi Serangan Pada Snort

Pada Gambar 4.8 menjelaskan suatu tempat penyimpanan paket *data* serangan dari hasil identifikasi serangan pada *snort* di dalam *var/log/snort*

6. Evaluasi Bug Scipt

Pada tahap ini penulis memberikan solusi untuk perbaikan *bug* pada *website* mengenai serangan *SQL Injection* yaitu :

1. Gunakan *Prepared Statement* pada *SQL Query* anda. *Prepared statements* adalah sebuah fitur yang disediakan oleh sebuah *Database Management System (DBMS)* seperti *MySQL, PostgreSQL, Oracle, SQLite, IBM, Firebird, DBLib*, dan lain-lain, dimana kita bisa mengirim *query* secara terpisah dari *query* utama dan *variable* pendukungnya. *Prepared Statement* adalah salah satu cara untuk melakukan komunikasi ke *database* yang dikatakan cukup efektif untuk mencegah *SQL Injection*. Tujuannya, agar *query* menjadi lebih aman dan cepat (jika perintah yang sama akan digunakan beberapa kali). Menggunakan *prepared statement* ini juga sangat disarankan ketika perlu melakukan banyak *query* supaya tidak terlalu memberatkan sistem.

Contoh *script Prepared Statement* pada *SQL Query :*

PHP

```php
<?php
$db_host = "localhost";
$db_name = "admin"; // database name
$db_user = "baru"; // datebase user
$db_pass = "123456"; // database password

$koneksi = "mysql:host=$db_host;dbname=$db_name";

try
{
    $db = new PDO($koneksi, $db_user, $db_pass);
}
catch (exception $e)
{
    echo "error";
    exit();
}
$id = $_GET['id'];
$sql = "SELECT * FROM pengguna WHERE id = $id";
$stmt = $db->prepare($sql);
$stmt->execute();
$objek = $stmt->fetchObject();
echo $objek->nama;

?>
```

Gambar 4.10 Tampilan *Script Prepared Statement* pada *SQL Query*

2. Gunakan *filter* pada *code PHP* anda.

Sebuah *filter PHP* digunakan untuk memvalidasi dan menyaring *data* yang berasal dari sumber yang tidak aman. *Filter* ini tentunya berguna untuk menyaring tipe *input* yang sedang dimasukkan oleh pengguna dan pastinya fitur *Filter* pada *php* ini juga bisa berguna untuk meminilisir serangan *SQL Injection* pada *parameter/website* anda. berikut contoh codenya:

PHP

```php
<?php
$id = filter_var($_POST['id'], FILTER_VALIDATE_INT); // Filter ini berguna untuk mefilter tipe
//data integer
$nama = filter_var($_POST['nama'], FILTER_SANITIZE_STRING);//untuk tipe string
?>
```

Gambar 4.11 Tampilan *Code Script Filter Php*

3. Gunakan *htaccess* untuk *memfilter Query HTTP* pada *web server* anda. Metode ini biasanya digunakan oleh plugin keamanan *Wordpress* berikut contoh codenya:

PHP

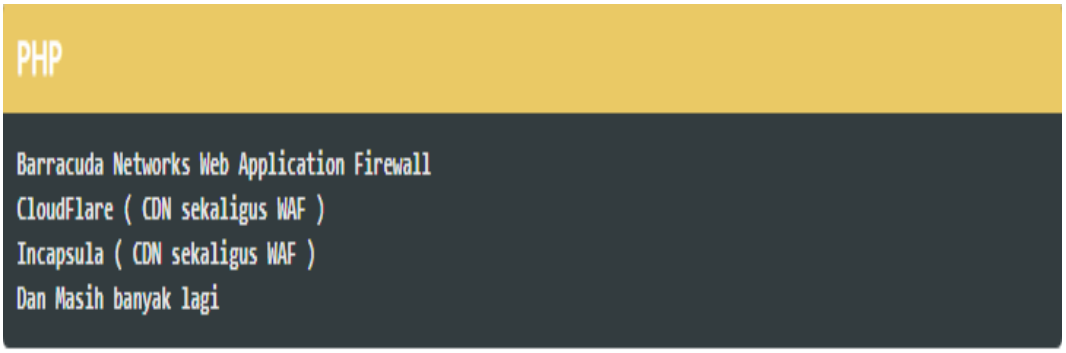
PLAIN

```
ServerSignature Off

Options -Indexes

RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^(HEAD|TRACE|DELETE|TRACK|DEBUG) [NC]
RewriteRule ^(.*)$ - [F,L]
RewriteCond %{REQUEST_URI} (timthumb\.php|phpthumb\.php|thumb\.php|thumbs\.php) [NC]
RewriteRule . - [S=1]
RewriteCond %{HTTP_USER_AGENT} (libwww-perl|wget|python|nikto|curl|scan|java|winhttp|clshttp|loader) [NC,(
RewriteCond %{HTTP_USER_AGENT} (<|>|'|%0A|%0D|%27|%3C|%3E|) [NC,OR]

RewriteCond %{HTTP_USER_AGENT} (;|<|>|'|"|\)|\(|%0A|%0D|%22|%27|%28|%3C|%3E|).*(libwww-perl|wget|python|ni

RewriteCond %{THE_REQUEST} \?\ HTTP/ [NC,OR]

RewriteCond %{THE_REQUEST} \/\*\ HTTP/ [NC,OR]

RewriteCond %{THE_REQUEST} etc/passwd [NC,OR]

RewriteCond %{THE_REQUEST} cgi-bin [NC,OR]

RewriteCond %{THE_REQUEST} (%0A|%0D) [NC,OR]

RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=http:// [OR]

RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=(\.\.//?)+ [OR]

RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=/([a-z0-9_.]//?)+ [NC,OR]

RewriteCond %{QUERY_STRING} \=PHP[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12} [NC,OR]
```

```
RewriteCond %{QUERY_STRING} (\.\./|\.\.) [OR]
RewriteCond %{QUERY_STRING} ftp\: [NC,OR]
RewriteCond %{QUERY_STRING} http\: [NC,OR]
RewriteCond %{QUERY_STRING} https\: [NC,OR]
RewriteCond %{QUERY_STRING} \=\|w\| [NC,OR]
RewriteCond %{QUERY_STRING} ^(.*)/self/(.*)$ [NC,OR]
RewriteCond %{QUERY_STRING} ^(.*)cPath=http://(.*)$ [NC,OR]
RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} (<|%3C)([^s]*s)+cript.*(>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} (\<|%3C).*iframe.*(\>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} (<|%3C)([^i]*i)+frame.*(>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [NC,OR]
RewriteCond %{QUERY_STRING} base64_(en|de)code[^(]*\([^)]*\) [NC,OR]
RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [OR]
RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2}) [OR]
RewriteCond %{QUERY_STRING} ^.*(\[|\]|\(|\)|<|>).* [NC,OR]
RewriteCond %{QUERY_STRING} (NULL|OUTFILE|LOAD_FILE) [OR]
RewriteCond %{QUERY_STRING} (\./|\../|\.../)+(motd|etc|bin) [NC,OR]
RewriteCond %{QUERY_STRING} (localhost|loopback|127\.0\.0\.1) [NC,OR]
RewriteCond %{QUERY_STRING} (<|>|'|%0A|%0D|%27|%3C|%3E|) [NC,OR]
RewriteCond %{QUERY_STRING} concat[^\(]*\( [NC,OR]
RewriteCond %{QUERY_STRING} union([^s]*s)+elect [NC,OR]
RewriteCond %{QUERY_STRING} union([^a]*a)+ll([^s]*s)+elect [NC,OR]
RewriteCond %{QUERY_STRING} (;|<|>|'|"|\)|%0A|%0D|%22|%27|%3C|%3E|).*(/\*|union|select|insert|drop|delete
RewriteCond %{QUERY_STRING} (sp_executesql) [NC]
RewriteRule ^(.*)$ - [F,L]
```

Gambar 4.12 Tampilan code *htaccess* untuk *memfilter Query HTTP*

4. Pasang WAF (Web Application Firewall) Pada *Web Server* Anda. *Firewall* ini adalah sebuah aplikasi keamanan untuk *web server* berada pada Layer 7 OSI. WAF ini di standari oleh OWASP (Open Web Application Security Project), merupakan kebijakan keamanan yang di posisiskan antara aplikasi *web* dan *end user.* Berikut beberapa contoh *code script* WAF yang dapat diandalkan untuk menjaga *Web* anda dari serangan *Hacker:*



Gambar 4.13 Tampilan *Web Application Firewall (WAF)*

7. Pencegahan Dengan Blokir Mac Filter Firewall

Pada tahap ini penulis memberikan solusi keamanan untuk memblokir *mac filter firewall* pada serangan *SQL Injection* ke website yaitu :

Blokir adalah aksi yang diambil untuk menghentikan orang-orang tertentu yang mengakses informasi. *Firewall* merupakan fitur yang biasanya banyak digunakan untuk melakukan filtering akses ***(****Filter Rule****)***, *Forwarding (NAT),* dan juga untuk menandai koneksi maupun paket dari trafik data yang melewati router (*Mangle*). Terdapat sebuah *parameter* utama pada *rule* di fitur *firewall* ini yaitu *Chain*. Parameter ini memiliki kegunaan untuk menetukan jenis trafik yang akan di-manage pada fitur *firewall* dan setiap fungsi pada *firewall* seperti *Filter Rule, NAT, Mangle* memiliki *opsi chain* yang berbeda.

Pengisian *parameter chain* pada dasarnya mengacu pada skema *Traffic Flow* dari *Router*. Jadi kita harus mengenali terlebih dahulu jenis *trafik* yang akan kita *manage* menggunakan *firewall. chain* bisa dianalogikan sebagai tempat admin mencegat sebuah trafik, kemudian melakukan *firewalling* sesuai kebutuhan

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil Analisa Serangan *SQL Injection* Pada *Web Server* menggunakan *Intruction Detection System* (IDS), maka didapat beberapa kesimpulan seperti berikut:

1. *System snort* yang dibagun dapat mengidentifikasi serangan *SQL Injection* dengan *type* serangan *attempted-admin* (Attempted Administrator Privilege Gain) *Priority:1* dan *snort* menggunakan *type snort Network Interuction Detection System* (NIDS) dengan mengunakan *Rule* tambahan yang dapat kita iformasikan paket-paket serangan apa saja yang harus teridentifikasi didalam *snort*.
2. Efek pada serangan *SQL Injection* berdampak pada kerugian terhadap sebuah informasi data yang berada dalam sebuah *database*, yang seketika *attacker* kapan saja dapat mengambil informasi data yang dibutuhkan *attacker* dalam *database* dan mensalahgunakannya.

5.2. Saran

Berikut adalah saran dari penulis agar analisa serangan *SQL Injection* pada *Web Server* menggunakan *Intruction Detection System* (IDS) ini dapat bermanfaat dan dikembangkan menjadi lebih baik lagi :

1. *System Web Server* hanya dapat aman dari serangan *SQL Injection* apabila seorang pemilik *Web Server* membangun atau menggunakan *system* keamanan yang benar-benar aman .
2. Cara terbaik untuk mencegah serangan *SQL Injection* adalah dengan memperbaiki *bug* yang rentan pada *website* yang berupa variable anti *SQL Injection*, dan agar lebih aman lagi kita dapat membagun sebuah *system* keamanan berupa WAF dan *Blokir Mac Filter Firewall*.

Daftar Pustaka

arrissetiawan. (2013). *No Title*. SELASA, 11 JUNI. http://arrissetiawan.blogspot.com/2013/06/metode-serangan-sql-injection-dan.html

Analisis Forensik Serangan SQL Injection dan DoS Menggunakan Instrution Detection System Pada Server Berbasis Lokal

Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). Jurnal Media Informatika Budidarma, 2(2).

Bhatt, D. (2018). Modern day penetration testing distribution open source platform-Kali Linux-study paper. *International Journal of Scientific and Technology Research*, *7*(4), 233–237.

Dewi, E. K. (2017). Analisis Log Snort Menggunakan Network Forensic. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *2*(2), 72–79. https://doi.org/10.29100/jipi.v2i2.370

Hadi, S. (2016). *Implementasi Network Intrusion Detection System pada Sistem Smart Identification*. *2*(3), 1171–1176.

Hamdani, H., Tharo, Z., & Anisah, S. (2019, May). PERBANDINGAN PERFORMANSI PEMBANGKIT LISTRIK TENAGA SURYA ANTARA DAERAH PEGUNUNGAN DENGAN DAERAH PESISIR. In Seminar Nasional Teknik (SEMNASTEK) UISU (Vol. 2, No. 1, pp. 190-195).

Khairul, K., IlhamiArsyah, U., Wijaya, R. F., & Utomo, R. B. (2018, September). IMPLEMENTASI AUGMENTED REALITY SEBAGAI MEDIA PROMOSI PENJUALAN RUMAH. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 429-434).

Khamphakdee, N., Benjamas, N., & Saiyod, S. (2015). Improving intrusion detection system based on snort rules for network probe attacks detection with association rules technique of data mining. *Journal of ICT Research and Applications*, *8*(3), 234–250. https://doi.org/10.5614/itbj.ict.res.appl.2015.8.3.4

Kholid, M. U. (n.d.). *Perancangan Sistem*. *3*, 38–44. http://www.academia.edu/7511410/Perancangan_Sistem_Menurut_Jogiyanto_H

Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. Jurnal Teknik dan Informatika, 5(2), 13-19.

Muttaqin, A., Akbar, S. R., & Brawijaya, U. (2014). Web Server Embedded System. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, *1*(1), 50–54.

Nanang, H., Fuadi, A., & Farhanah, N. (2008). *Pengembangan Aplikasi Remote*. *2*(3), 1–8.

Ngatmono, D., Riasti, B. K., & Sasongko, D. (2015). Membangun Sistem Operasi Mandiri Berbasis Open Source Dengan Metode Remaster. *Indonesian Journal on Networking and Security*, *4*(3), 39–47.

Nugroho, D. A., Rochim, A. F., & Widianto, E. D. (2015). Perancangan dan Implementasi Instrusion Detection System di Jaringan Universitas Diponegoro. *Jurnal Teknologi Dan Sistem Komputer*, *3*(2), 171. https://doi.org/10.14710/jtsiskom.3.2.2015.171-178

No Title, (2018). https://otanaha.web.id/tekno/2018/12/28/mengenal-sql-injection/

Pamungkas, C. ajika. (2016). Manajemen bandwith menggunakan mikrotik routerboard di politeknik indonusa surakarta. *Jurnal Informa*, *1*(3), 3–8.

Pi, R. (2019). *Implementasi IPTables untuk Packet Filtering Firewall*. *6*(1), 61–66.

Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. Int. J. Eng. Technol., 7(2.13), 345-347.

Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., ... & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. In Journal of Physics: Conference Series (Vol. 1028, No. 1, p. 012052). IOP Publishing.

Rahmaniar, R. (2019). Model FLASH-NR Pada Analisis Sistem Tenaga Listrik (Doctoral dissertation, Universitas Negeri Padang).

Rossanty, Y., Aryza, S., Nasution, M. D. T. P., & Siahaan, A. P. U. (2018). Design Service of QFC And SPC Methods in the Process Performance Potential Gain and Customers Value in a Company. Int. J. Civ. Eng. Technol, 9(6), 820-829.

S, G. R. (2015). No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, *16*(2), 39–55. https://doi.org/10.1377/hlthaff.2013.0625

Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. Int. J. Sci. Res. Sci. Technol, 3(6), 470-473.

Siahaan, A. P. U., Ikhwan, A., & Aryza, S. (2018). A Novelty of Data Mining for Promoting Education based on FP-Growth Algorithm.

Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.

Sidik, A. P., Efendi, S., & Suherman, S. (2019, June). Improving One-Time Pad Algorithm on Shamir's Three-Pass Protocol Scheme by Using RSA and ElGamal Algorithms. In Journal of Physics: Conference Series (Vol. 1235, No. 1, p. 012007). IOP Publishing.

Sistem, J., & Dan Manajemen, I. (2018). Analisis Network Security Snort Menggunakan Metode Intrusion Detection System (Ids) Untuk Optimasi Keamanan Jaringan Komputer. *JURSIMA Jurnal*, *6*(1).

Sitorus, Z. (2018). Kebutuhan Web Service untuk Sinkronisasi Data Antar Sistem Informasi dalam Universitas. Jurnal Teknik dan Informatika, 5(2), 87-90.

Subrata, K. (2015). *Flowchart, Pendahuluan Membuat, Pedoman-pedoman Dalam Flowchart*. 1–13.

Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 100-109.

Tasril, V., Wijaya, R. F., & Widya, R. (2019). APLIKASI PINTAR BELAJAR BIMBINGAN DAN KONSELING UNTUK SISWA SMA BERBASIS MACROMEDIA FLASH. Jurnal Informasi Komputer Logika, 1(3).

Universitas, S., Mada, G., & Mada, G. (2013). Analisis Forensik Jaringan Studi Kasus Serangan SQL Injection pada Server Universitas Gadjah Mada. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *6*(2). https://doi.org/10.22146/ijccs.2157

VARIANTO, E., & MOHAMMAD BADRUL. (2015). Implementasi Virtual Private Network Dan Proxy Server Menggunakan Clear Os Pada Pt.Valdo International. *Jurnal Teknik Komputer Amik Bsi*, *1*(1), 55–56.

Wamiliana, Wardhana, W., & Kharismaldie, F. (2013). Pembangunan Sistem Operasi Berbasis Linux Menggunakan Metode Linux From Scratch. *Jurnal Komputasi*, *1*(2), 30–37. http://jurnal.fmipa.unila.ac.id/index.php/komputasiHal.30dari94