ANALISA KEAMANAN SERVER PADA LOGIN PAGE WEB SERVER DENGAN ENKRIPSI SHA 1 DARI SERANGAN SQL INJECTION MENGGUNAKAN SISTEM OPERASI KALI LINUX DI LKP MULTI LOGIKA BINJAI

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA	**: WENI DWI FATMA**
NPM	**: 1514370062**
PROGRAM STUDI	**: SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

ABSTRAK

WENI DWI FATMA

ANALISA KEAMANAN SERVER PADA LOGIN PAGE WEBSERVER DENGAN ENKRIPSI SHA 1 DARI SERANGAN SQL INJECTION MENGGUNAKANSYSTEMOPERASI KALI LINUX DI LKP MULTI LOGIKA BINJAI

2020

Semakin meningkat pertumbuhan layanan informasi maka semakin tinggi pula tingkat kerentanan keamanan dari suatu sumber informasi. Salah satu cara untuk mengakses internet adalah menggunakan aplikasi *web*. Tampilan *web* yang interaktif menyebabkan pengguna dapat memakai web dengan mudah. Namun dibalik kemudahan penggunaan *web*, ada faktor lain yang kurang diperhatikan yaitu keamanan yang merupakan aspek penting dalam aplikasi *web*. Ancaman yang menempati urutan teratas yang dapat dilakukan pada aplikasi *web* adalah *injection*. Salah satu *injection* yang paling umum dilakukan adalah pada *databaseSQL*. *SQL injection* merupakan salah satu tindakan yang mencurigakan yang memanfaatkan celah keamanan pada *database SQL* dengan menyisipkan *query* ilegal yang bertujuan untuk *bypass login*, memanipulasidata dan merusak *database*. Melalui tulisan ini disajikan penelitian yang dilakukan secara eksperimen yang membahas tentang serangan SQL *Injection*. Pengamanan sebuah *server* pada *login page web server* dengan menggunaakan enkripsi SHA 1 yang merupakan pintu pertama akses yang seharusnya memiliki pertahanan yang cukup. Kemudian dilakukan pengujian dengan menggunakan system operasi kali linux. untuk mendapatkan solusi terhadap permasalahan tersebut.

Kata Kunci : *Sql Injection, Page Web Server, SHA 1, Kali Linux*

KATA PENGANTAR

Bismillahirrohmanirrohim

Syukur alhamdulillah penulis panjatkan puji syukur kehadirat Allah SWT, karena atas berkat dan rahmat karunia – Nya Penulis dapat menyelesaikan penyusunan skripsi ini dengan judul **"ANALISA KEAMANAN SERVER PADA LOGIN PAGE WEB SERVER DENGAN ENKRIPSI SHA 1 DARI SERANGAN SQL INJECTION MENGGUNAKAN SYSTEM OPERASI KALI LINUX DI LKP MULTI LOGIKA BINJAI ".**

Penyusunan skripsi ini adalah salah satu syarat yang harus di penuhi untuk menyelesaikan pendidikan S-1 pada studi Sistem Komputer Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi Medan.

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

1. **Kedua Orang Tua Bapak Wagimun Dan Ibu Misnah** yang telah memberikan kasih sayang, dorongan, do'a, nasihat dan motivasi dari segi moril dan materil selama selama penulis penempuh pendidikan di Universitas Pembangunan Panca Budi Medan.
2. **Bapak Dr H. Muhammad Isa Indrawan, S.E., M.M,** selaku Rektor Universitas Pembangunan Panca Budi Medan.
3. **Bapak Hamdani, S.T., M.T,** selaku Dekan Fakultas Sains & Tekhnologi Universitas Pembangunan Panca Budi Medan.
4. **Bapak Eko Hariyanto, S.Kom., M.Kom.,** Selaku Ketua Program Studi Sistem Komputer Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi Medan
5. **Bapak Dr Muhammad Iqbal, S.Kom.,M.Kom,** selaku Dosen Pembimbing I (Satu) Tugas Akhir yang telah banyak memberikan bimbingan kepada penulis.
6. **Bapak Dian Kurnia , S.Kom., M.Kom,** selaku Dosen Pembimbing II (Dua) Tugas Akhir yang telah banyak memberikan bimbingan kepada penulis.
7. **Bapak Hefri Syafrudin** Selaku Pimpinan LKP Multi Logika Binjai.
8. **Seluruh Staf Pegawai** Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi Medan yang telah banyak membantu dalam kelancaran seluruh aktivitas perkuliahan .
9. **Seluruh Dosen-dosen** program studi sistem komputer Universitas Pembangunan Panca Budi Medan.
10. **Seluruh staf** LKP Multi Logika Binjai.
11. **Sahabat – Sahabat** Tersayang Dewi Maduma Harahap, Amalia Ramadhani Sembiring, Ike Sri Cahyati Pulungan, Legiati dan yang lainnya yang selalu memberi dukungan dan semangat kepada penulis.

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 sifatnya membangun dari pembaca untuk penyempurnaan isi Skripsi ini.

Medan, 24 Januari 2020

Penulis,

Weni Dwi Fatma
NPM : 1514370062

DAFTAR ISI

Halaman

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Menjelaskan dengan perkembangan teknologi internet saat ini memungkinkan pertukaran informasi seperti ilmu pengetahuan, hiburan, berita dan jenis informasi lain secara *real-time*. Faktor kemudahan dan kenyamanan ini menyebabkan internet menjadi media informasi yang paling banyak digunakan saat ini (Irawan, Pramukantoro, & Kusyanti, 2018).

Kondisi atau perkembangan Multi Logika Binjai saat ini adalah kurang nya tingkat keamanan untuk mengakses sebuah jaringan adanya pengaruh serangan *SQL Injection* serangan ini dapat dilakukan melalui input *user* maupun url yang dapat masuk dan merusak sistem multi logika, cara mengetahui ancaman yang terjadi pada server adanya kejahatan yang dilakukan dengan membuat gangguaan, perusakan atau penghancuraan terhadap data atau sistem jaringan komputer.

Salah satu cara untuk mengakses internet adalah menggunakan aplikasi *web*. Tampilan *web* yang interaktif menyebabkan pengguna dapat memakai web dengan mudah. Namun dibalik kemudahan penggunaan *web*, ada faktor lain yang kurang diperhatikan yaitu keamanan yang merupakan aspek penting dalam aplikasi *web*. Ancaman yang menempati urutan teratas yang dapat dilakukan pada aplikasi *web* adalah *injection*. Salah satu *injection* yang paling umum dilakukan adalah pada *database SQL*. *SQL injection* merupakan salah satu tindakan yang mencurigakan yang memanfaatkan celah keamanan pada *database SQL* dengan

Menyisipkan *query* ilegal yang bertujuan untuk *bypass login*, memanipulasi data dan merusak *database*. Untuk mengurangi ancaman ini, dibutuhkan mekanisme pertahanan yang dapat melindungi sistem dari *injection*. Metode Pencegahan yang sering digunakan saat ini adalah menerapkan *rule-rule* yang bertujuan untuk mencegah tindakan yang membahayakan *database SQL*.

Menjelaskan pengujian keamanan sistem komputer adalah suatu kegiatan yang dilakukan untuk menemukan celah keamanan yang terdapat pada sistem tersebut . Hasil pengujian ini dapat digunakan untuk memperbaiki sisi keamanan dari sistem untuk melindungi data-data dari serangan dan atau pencurian yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab. Selama ini, untuk melakukan pengujian keamanan membutuhkan beberapa aplikasi yang dijalankan pada suatu sistem operasi. Dua contoh sistem operasi yang tersebut adalah kali linux dan SHA 1, kedua sistem operasi biasa digunakan oleh para penguji system keamanan komputer. Dalam hal ini, penguji adalah orang yang melakukan penetrasi tentang atau pengujian sistem komputer dan menemukan celah keamanan pada sistem tersebut (Ar, Prayudi, & Yudha, 2018).

Dengan pertimbangan diatas dan pribadi penulis yang ingin meneliti dan menulis secara ilmiah sebagai tugas akhir dengan judul : "**ANALISA KEAMANAN SERVER PADA LOGIN PAGE WEB SERVER DENGAN ENKRIPSI SHA 1 DARI SERANGAN SQL INJECTION MENGGUNAKAN SYSTEM OPERASI KALI LINUX DI LKP MULTI LOGIKA BINJAI ".**

1.2 Rumusan masalah

Berdasarkan latar belakang di atas, maka dapat ditarik beberapa rumusan masalah yaitu :

1) Bagaimana menganalisa *page login web* yang telah diterapkan keamanan enkripsi SHA 1 didalamnya dan bagaimana pengaruh ancaman dari *SQL Injection* pada *web server* .

1.3 Batasan Masalah

Agar pembahasan masalah tidak menyimpang dari tujuan penelitian, maka berikut adalah beberapa batasan yang perlu dibuat, yaitu:

1. Hanya membahas enkripsi SHA 1 pada *page login web.*
2. *Web server* dalam kondisi *offline* pengujian.
3. Penguji *SQL Injection* dijalankan menggunakan jaringan wifi.
4. *Web server* dibangun menggunakan sistem operasi kali linux.
5. Database yang digunakan adalah *MySQL* dengan aplikasi *apache*.

1.4 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Menganalisa *page login web server* yang terenkripsi SHA 1 menggunakan *SQL Injection* dengan memanfaatkan sistem operasi kali linux.
2. Membangun *web server* menggunakan sistem operasi kali linux.
3. Mengetahui cara mengantisifasi dari serangan *SQL Injection*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan agar dapat memberikan manfaat sebagai berikut :

1. SHA 1 dapat menanggulangi adanya suatu serangan dari *SQL Injection*.
2. Serangan *SQL Injection* dapat diatasi dengan menggunakan sistem operasi Kali linux.
3. Dengan adanya *page login web server* dapat mempermudah untuk menganalisa.

BAB II

LANDASAN TEORI

2.1 Keamanan Jaringan

Menjelaskan keamanan jaringan secara umum adalah komputer yang terhubung ke *network*, mempunyai ancaman keamanan lebih besar dari pada komputer yang tidak terhubung kemana – mana . Dengan pengendalian yang teliti, resiko tersebut dapat dikurangi, namun *network security* biasanya bertentangan dengan *network access,* dimana bila *network access* semakin mudah, maka *network security* semakin rawan, begitu pula sebaliknya. Keamanan jaringan (*network security*) dalam jaringan komputer sangat penting dilakukan untuk memonitor akses jaringan dan mencegah penyalahgunaan sumber daya jaringan yang tidak sah. Tugas keamanan jaringan dikontrol oleh administrator jaringan (Studi et al., 2017).

Secara umum terdapat 3 hal dalam konsep keamanan jaringan, yaitu :

1. Resiko atau tingkat bahaya (*risk*)

   Menyatakan seberapa besar kemungkinan dimana penyusup (*instruder*) berhasil mengakses komputer dalam suatu jaringan.

2. Ancaman (*threat*)

   Menyatakan sebuah ancaman yang datang dari seseorang yang mempunyai keinginan untuk memperoleh akses ilegal ke dalam suatu jaringan komputer seolah-olah mempunyai otoritas terhadap jaringan komputer.

3. Kerapuhan sistem (*vulnerability*)

Menyatakan seberapa kuat sistem keamanan suatu jaringan komputer yang dimiliki dari seseorang dari luar sistem yang berusaha memperoleh akses ilegal terhadap jaringan komputer tersebut (Dasar & Jaringan, 2017).

Keamanan sendiri menyangkut 3 elemen dasar yaitu :

1. Keamanan jaringan (*network security*)
2. Keamanan aplikasi (*application security*)
3. Keamanan komputer (*computer security*)

2.2 *Virtual Server*

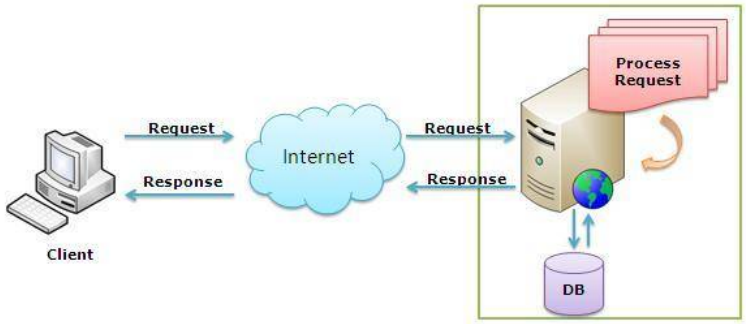
Virtual Machine adalah program sebuah sistem komputer yang dapat menjalankan banyak sistem operasi diatas sistem operasi lain dan digunakan untuk menjalankan aplikasi untuk sistem operasi lainnya. *Virtual machine* akan membuat suatu sistem *virtual* yang nantinya dapat diisi sebuah sistem operasi yang tidak berhubungan dengan sistem operasi utamanya (Soepomo, 2014).

2.3 Web *Server*

Menjelaskan *server* dalam dunia komputer adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung dengan *prosesor* yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan atau *network operating system*. *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (*printer*),dan memberikan akses kepada *workstation* anggota jaringan. Umumnya, di atas sistem

operasi *server* terdapat aplikasi-aplikasi yang menggunakan arsitektur klien/ *server*. Contoh dari aplikasi ini adalah DHCP Server, Mail Server, HTTP Server, FTP Server, DNS Server dan lain sebagainya.

Web Server merupakan salah satu mesin yang dimana tempat *software* atau aplikasi beroperasi dalam mendistribusikan *web page* ke user/pengguna, Protokol untuk mentransfer atau memindahkan berkas yang diminta oleh pengguna melalui protokol komunikasi tertentu. Oleh karena dalam satu halaman *web* biasanya terdiri dari berbagai macam jenis berkas seperti gambar, video, teks, audio, file dan lain sebagainya, maka pemanfaatan *web server* berfungsi juga mentransfer aspek pemberkasan dalam halaman tersebut, termasuk teks, gambar, video, audio, file dan sebagainya (Tedyyana, 2016).



Gambar 2.1 Web Server

Sumber: (Nasional, Informasi, Rahmatulloh, & Msn, 2017)

2.4 Ubuntu *Server*

Ubuntu merupakan salah satu distribusi *Linux* yang berbasiskan Debian dan didistribusikan sebagai *software* bebas. nama Ubuntu berasal dari filosofi dari Afrika Selatan yang berarti “Kemanusiaan kepada sesama”. Ubuntu didesain untuk kepentingan penggunaan personal, namun versi *server* Ubuntu juga

tersedia, dan telah dipakai secara luas. Proyek Ubuntu resmi disponspori oleh *Canonical* Ltd. yang merupakan sebuah perusahaan yang dimiliki oleh pengusaha Afrika Selatan *Mark Shuttleworth*. Tujuan dari distribusi *Linux* Ubuntu adalah membawa semangat yang terkandung di dalam Filosofi Ubuntu ke dalam dunia perangkat lunak. Ubuntu adalah sistem operasi lengkap berbasis *Linux*, tersedia secara bebas dan mempunyai dukungan baik yang berasal dari komunitas maupun tenaga ahli profesional dan Ubuntu juga bersifat *Open Source* (Soepomo, 2014).

2.5 Enkripsi dan Dekripsi

Menjelaskan proses menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*) atau *enciphering* (standar nama menurut ISO 7498-2). Proses mengembalikan *ciphertext* menjadi *plaintext*-nya disebut dekripsi (*decryption*) atau *deciphering* (standard nama menurut ISO 7498-2) .

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). Enkripsi dapat diartikan sebagai kode atau *cipher*. Proses yang dilakukan untuk mengamankan pesan (yang disebut *plainteks*) menjadi pesan yang tersembunyi (disebut *chiperteks*) adalah enkripsi (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext* disebut deskripsi (*decryption*) (Simargolang, 2017).

2.6 *Secure hash algoritma*-1 (SHA -1)

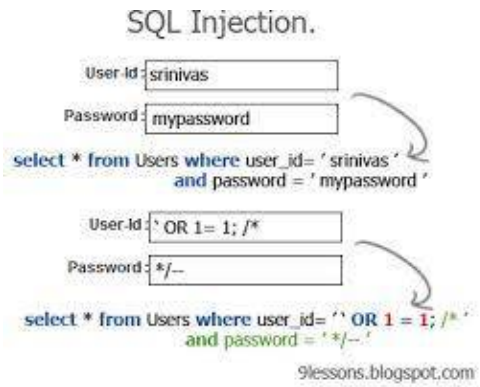
Menjelaskan SHA dikembangkan oleh *National Institute of Standards and Technology* (NIST) dan dipublikasikan sebagai Federal *Information Processing Standards* (FIPS 180) pada tahun 1993. *Secure Hash Standard* (SHS) menspesifikasikan SHA-1 untuk menghitung nilai hash dari sebuah pesan atau file. SHA-1 memiliki panjang pesan maksimal 264 *bits* dan memiliki keluaran sebesar 160 bits yang dinamakan *message digest* atau hash code. *Message digest* tersebut dapat digunakan sebagai masukan untuk Digital *Signature Algorithm* (DSA), yang digunakan untuk menghasilkan *signature* untuk memverifikasi pesan tersebut. Pada saat ditemukan kelemahan pada algoritma SHA-0, berbagai revisi dan perbaikan dilakukan untuk membuat suatu algoritma yang lebih baik. Hasil dari perbaikan tersebut diterbitkan pada tahun 1995 dan dijadikan acuan untuk pembuatan algoritma SHA-1. Algoritma SHA-1 merupakan revisi teknis dari algoritma SHA. Algoritma SHA-1 dapat dikatakan aman karena proses perhitungannya tidak memungkinkan untuk menemukan pesan yang sebenarnya dari *message digest* yang dihasilkan. Setiap perubahan yang terjadi pada pesan saat perjalanan akan menghasilkan *message digest* yang berbeda. Algoritma SHA-1 berbasis pada algoritma MD4 dan rancangannya sangatlah mirip terhadap algoritma tersebut (Kurniawan, Kusyanti, & Nurwarsito, 2017).

2.7 *SQL Injection*

Menjelaskan *SQL injection* merupakan teknik untuk mengeksploitasi sebuah aplikasi web memakai data yang diberikan atau disisipkan dalam *query SQL*. Cara

kerja *SQL injection* dengan cara memasukan *query SQL* atau perintah (*command*) sebagai input yang dimungkinkan melalui halaman *web* atau *command prompt*. Halaman *web* akan mengambil parameter dari user lalu membuat *query SQL* masuk kedalam database. Dengan demikian, *SQL injection* dapat pula dikatakan sebagai suatu kegiatan yang menipu *query* dari database, sehingga seseorang yang tidak ter-otentikasi dapat mengetahui dan mendapatkan informasi yang terdapat pada database (Irawan et al., 2018).

Menjelaskan sebuah kerentanan yang menyebabkan seorang penyerang memiliki kemampuan untuk mempengaruhi *query SQL* yang dikirimkan melalui aplikasi ke *database*. Dengan kemampuan tersebut seorang penyerang dapat mempengaruhi *syntax SQL*, kekuatan, fleksibilitas dari *database* pendukung fungsional dan mempengaruhi fungsi sistem operasi yang dialokasikan untuk *database*. *SQL Injection* tidak hanya mempengaruhi aplikasi *web* tapi juga semua program lain yang menggunakan kalimat *SQL*. Semua program yang menggunakan input dinamis dari luar (*untrusted*) dapat terserang oleh SQL *(Yulianingsih, 2017).*



Gambar 2.2 SQL Injection

Sumber: (Sandi, Putra, & Selatan, 2017)

2.8 Klasifikasi SQL *injection*

Menjelaskan SQL injection merupakan serangan yang ditujukan pada *database* dengan menginputkan *query* tanpa validasi atau ilegal. Untuk melakukan *injection*, penyerang dapat melakukan nya dengan beberapa teknik penulisan (Irawan et al., 2018) :

1. *Tautologies*

   Penyerang menginputkan *query* yang hasilnya selalu "True":

   Select * From Users where

   Username='admin' or 1=1 --' and Password=' ' ;

2. *UnionQueries*

   Penyerang menginputkan "*UNION QUERY*" untuk mendapatkan lebih banyak data:

   Select bookTitle, ISBN from books where

   bookID = 1 UNION Select "hack", balance

   from accounts where accNo = 3456 --;

3. *Piggyback Queries*

   Pengarang menginputkan tambahan *statement* yang mempengaruhi hasil *query*:

   Select * from users where username='';

   drop table accounts -- and password=''

4. *Malformed Queries*

   Penyerang menggunakan kerentanan pada hasil *error message* untuk mendapatkan informasi dari *database* :

Select * from books where

BookID=convert(unt,(select top 1 name

from sysobjects where xtype =’u‘)) ;

5. *Inference*

Jenis serangan ini bergantung pada response-time pada *web server* untuk mendapatkan informasi :

Select * from users where

Username=’hello’; select if(user()

Like ‘root@%’, benchmark(1000000,

Shal(‘test’)),password ‘false’); --‘ and

6. *Alternate Encoding*

Jenis serangan ini menggunakan kombinasi dari spesial karakter (seperti *quote,dash* dll) untuk melewati skema pertahanan:

Select * from books where

bookID=1;exec(char(0x730065006c00650063

007400200040004000760065007200730069006

f006e00);

2.9 SQLmap

SQLMap adalah aplikasi *open source* atau *tool* yang terdapat dalam kali linux. Aplikasi ini digunakan untuk mendeteksi dan mengekploitasi kerentanan aplikasi *web*, aplikasi imi mampu mengambil alih *server database*. Dengan menggunakan SQLMap penyerang atau tester dapat melakukan penyerangan pada

database SQL menjalankan perintah pada sistem operasi, , mengambil detail struktur *database,*Melihat atau menghapus data yang terdapat dalam *database* dan bahkan mengakses sistem *file* dari *server*. SQLMap mendukung enam teknik injeksi SQL *Boolean-based blind*, *Time –based blind*, *Error based,* UNION-*based*, *Inteferential*, dan *Out –of-band*. *Boolean – based blind* adalah teknik injeksi yang bergantung pada pengiriman perintah SQL ke *database* yang memaksa aplikasi untuk mengembalikan hasil yang berbeda . *Time –based* adalah teknik injeksi yang bergantung pada pengiriman perintah SQL ke *database* yang memaksa *database* untuk mengganggu waktu yang telah ditentukan sebelum merespons untuk menunjukkan kepada penyerang apakah hasil perintah tersebut benar atau salah. Error *–based* adalah teknik injeksi yang bergantung pada pesan kesalahan yang dikirim oleh *database* untuk mendapatkan informasi tentang struktur (Lika, Dwi, Halim, & Verdian, 2018).

2.10 Sistem Operasi berbasis *linux*

Menjelaskan *Linux* adalah sistem operasi berbasis *GNU/Linux* yang bersifat *Open Source* dan memiliki banyak varian seperti Debian, *Slackware, Open Suse, Archlinux, Redhat* dan sebagainnya. Walaupun sangat banyak varian *GNU/Linux* hanya menyediakan aplikasi yang sudah ditentukan yang mungkin kurang bermanfaat oeleh pengguna yang melakukan *remastering* untuk memenuhi kebutuhannya. *Remastering* adalah proses membuat sistem operasi baru dengan mengurangi atau menambahkan fitur – fiturnya dari distro *GNU/Linux* yang telah ada. Ada beberapa *GNU/Linux* hasil *remaster* dikhususkan untuk kebutuhan

tertentu diantarannya seperti *Ubuntu* studio yang dibuat untuk keperluan multimedia. *GNU/Linux sabily* yang dibuat untuk umat muslim dan *Backtrack*/Kali untuk kebutuhan *Penetration* testing.Tujuannya untuk mempermudah, mempercepat pemasangan karena kendala keterbatasan koneksi internet dan konfigurasi kebutuhan pemrograman pada *GNU/Linux*.

Linux adalah sebuah aplikasi atau program yang menggunakan kernel sebagai sistem operasi . *Script* pertama linux dirancang dan ditulis oleh seorang mahasiswa dari Finlandia bernama "*Linus Torvalds*" untuk Intel 80386 arsitektur. *Script* lain dari Linux yang tersedia di Internet pada tahun 1991. Setelah itu, banyak orang bermain peran penting dalam mengembangkan dan memperluas *Linux* di berbagai belahan dunia. Sistemnya, peralatan sistem dan pustakanya umumnya berasal dari sistem operasi GNU, yang diumumkan tahun 1983 oleh Richard Stallman (Harjono, Sarjana, & Ilmu, 2019)**.**

2.11 Kali Linux

Kali Linux adalah distribusi berlandasan distribusi Debian GNU/Linux untuk tujuan forensik digital dan di gunakan unntuk pengujian penetrasi yang dipelihara dan didanai oleh *Offensive Security*. Kali juga dikembangkan *oleh Offensive Security* sebagai penerus *BackTrack Linux* Kali menyediakan pengguna dengan mudah akses terhadap koleksi yang besar dan komprehensif untuk alat yang berhubungan dengan keamanan, termasuk port scanner untuk *password cracker*. Pembangunan kembali *Back Track Linux* secara sempurna, mengikuti sepenuhnya

kepada standar pengembangan debian. Semua infrastruktur baru telah dimasukkan ke dalam satu tempat, semua tools telah *direview* dan dikemas dan kami menggunakan Git untuk VCS nya (Muhammad Addy Rahmadani, Mochammad Fahru Rizal, 2017).

2.12 UML

Menjelaskan *Unified Modeling Language (UML)* adalah bahasa spesipikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

2.12.1 Tujuan atau fungsi dari penggunaan UML

Inilah beberapa tujuan atau fungsi dari penggunaan uml, yang diantaranya :

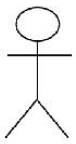
1. Dapat memberikan bahasa pemodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktek – praktek terbaik yang ada dalam pemodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.

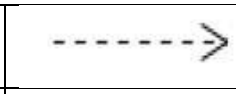
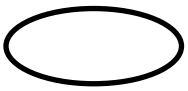
4. Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yaang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
6. Dapat menciptakan suatu bahasa pemodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

2.12.2 *Use Case Diagram*

Use Case Diagram Merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut.

Tabel 2.1 Simbol *Use Case* Diagram

No.	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mendiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (*ancestor*).

4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit.*
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antar objek satu dengan objek lainnya.
7		*Sistem*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use case*	*Deskripsi* dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemen (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber : (Windu Gata, 2016).

2.12.3 Diagram aktivitas (*Activity diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Tabel 2.2 Simbol Activity Diagram

No.	Gambar	Nama	Keterangan
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi.

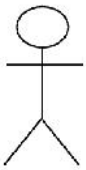
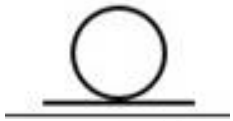
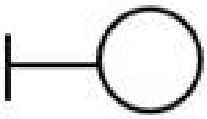
3		*Initial node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity final node*	Bagaimana objek dibentuk dan dihancurkan.
5		*Fork node*	Satu aliran pada tahap tertentu berubah menjadi beberapa aliran.

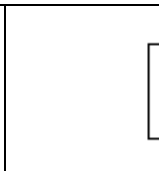
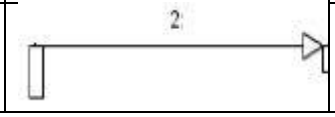
Sumber : (Windu Gata, 2016).

2.12.4 Diagram Urutan (*sequence Diagram*)

sequence Diagram Menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

Tabel 2.3 Simbol Sequence Diagram

No.	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Entity class*	Menggambarkan hubungan yang akan dilakukan.
3		*Boundary class*	Menggambarkan sebuah gambaran dari *foem*.
4		*Control class*	Menggambarkan penghubung antara *boundary* dengan tabel.

5		*A focus of control & a life line*	Menggambarkan tempat mulai dan berakhirnya *massage.*
6		*A massage*	Menggamabrkan pengiriman pesan.

Sumber : (Windu Gata, 2016).

2.12.5 Diagram kelas (*class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuaah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class Diagram secaara khas meliputi :

Kelas (*Class*), Relasi *Assosiations,Generalitation* dan *Aggregation* ,attribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yaang disebut dengan *multiplicity* atau *cardinality* (Windu Gata, 2016).

Tabel 2.4 Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		*Asosiasi / association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity.*
2.		Asosiasi berarah /*directed association*	Relas antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.*
3.		*Generalisasi*	Relas antar kelas dengan makna generalisasi-*spesialisasi* (umum khusus.
4.		Kebergantungan / *dependency*	Relasi antar kelas dengan makna kebergantungan antar kelas.
5.		*Agregasi / agregation*	Relasi antar kelas dengan makna semua-bagian (*whole-part*).

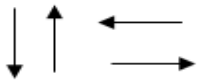
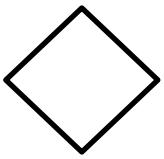
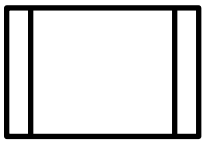
Sumber : (Windu Gata, 2016).

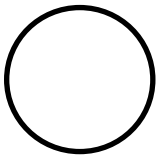
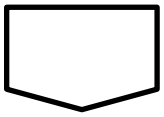
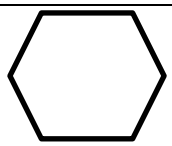
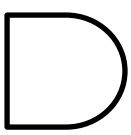
2.13 Flowchart

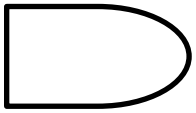
Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah,disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang berkerja dalam tim suatu proyek. *flowchart* membantu memahami urutan-urutan logika yaang rumit dan

panjang. *flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah (Nurmalina, 2017).

Tabel 2.5 Simbol-simbol Flowchart

No.	Simbol	Nama Simbol	Keterangan
1.		Terminal *Point* Symbol/ Simbol Titik Terminal	Menunjukkan permulaan (*start*) atau akhir (*stop*) dari suatu proses.
2.		*Flow Direction* Symbol/ Simbol Arus	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain (*conneting line*).
3.		*Processing* Symbol/ Simbol Proses	Simbol yang digunakan untuk menunjukkan kegiatan yang dilakukan oleh komputer.
4.		*Decision* Symbol/ Simbol Keputusan	Simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada.
No.	**Simbol**	**Nama Simbol**	**Keterangan**
5.		*Input-Output*/ Simbol Keluar-Masuk	Menunjukkan proses *input-output* yang terjadi tanpa bergantung dari jenis peralatannya.
6.		*Predefined* Process/ Simbol Proses	Simbol yang digunakan untuk mennujukkan pelaksanaan suatu bagian prosedur (sub-proses).

7.		*Connector (On-page)*	Simbol ini fungsinya adalah untuk menyederhanakan hubungan antar simbol yang letaknya berjauhan atau rumit bila dihubungkan dengan garis dalam satu halaman.
8.		*Connector (Off-page)*	Sama seperti *on-page connector*, hanya saja simbol ini digunakan untuk menghubungkan simbol dalam halaman berbeda.
9.		*Preparation* symbol/ Simbol Persiapan	Simbol yang digunakan untuk mempersiapkan penyimpanan didalam *storage*.
10.		Manual *Input* Symbol	Digunakan untuk menunjukkan *input* data secara manual menggunakan *online keyboard*.
11.		Manual *Operation* Symbol/ Simbol Kegiatan Manual	Digunakan untuk menunjukkan kegiatan/proses yang tidak dilakukan oleh komputer.
12.		Document Symbol	Simbol ini mengartikan *input* berasal dari dokumen dalam bentuk kertas, atau *output* yang perlu dicetak diatas kertas.
13.		*Multiple Documents*	Sama seperti document symbol hanya saja dokumen yg digunakan lebih dari satu dalam simbol ini.
14.		*Display Symbol*	Adalah 22simbol yang menyatakan penggunaan peralatan *output*, seperti layar *monitor*, *printer*, *plotter* dan lain sebagainya.

15.		*Delay Symbol*	Sesuai dengan namanya digunakan untuk menunjukkan proses *delay* (menunggu) yang perlu dilakukan.

Sumber : (Nurmalina, 2017).

2.14 Sejarah dan Profil Perusahaan

Pada tahun 2010, Bapak Hefri membuka sebuah usaha Jual Beli Komputer dengan nama "Orbit Com", yang saat itu melayani jasa pemeliharaan dan perbaikkan komputer untuk instansi, lembaga pendidikan dan kalangan umum. Kemudian pada tahun 2012 mengajukan ijin operasional Lembaga Kursus Pelatihan Komputer dan Teknik Jaringan dengan nama Multi Logika yang berlokasi di Jalan Danau Laut Tawar No. 6 Km. 19 Kel. Sumber Mulyo Rejo Kec. Binjai Timur.

Tahun 2016 Multi Logika dilakukan akreditasi oleh Lemsar.net. Program dimasa mendatang "Multi Logika" akan selalu meningkatkan kualitas, baik Lembaga, Pendidik, Tenaga Kependidikan maupun peserta didik dan menjalankan Visi dan Misi, Berikut ini adalah visi dan misi LKP Multi Logika Binjai:

a. Visi

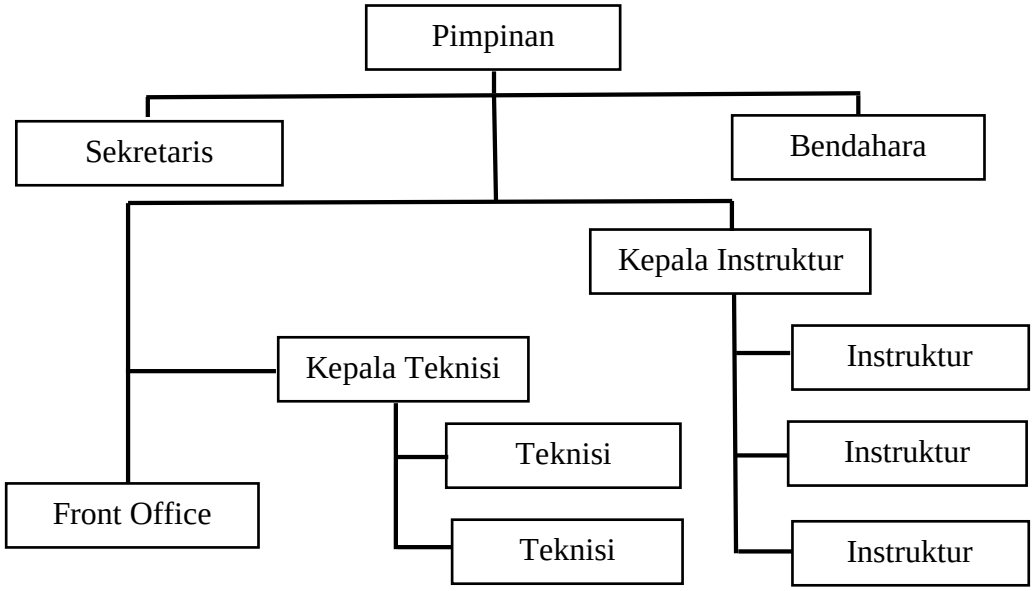
   Visi dari LKP Multi Logika Binjai adalah menjadi Lembaga Kursus dan Pelatihan yang unggul di tingkat Lokal maupun Nasional di bidang TI.

b. Misi

1) Menyelenggarakan program Life School yang sesuai dengan kebutuhan pasar.
2) Menghasilkan lulusan yang siap kerja maupun mandiri.

3) Membangun hubungan harmonis antar lembaga dengan masyarakat.

1. Struktur Organisasi LKP Multi Logika Binjai



Gambar 2.3 Struktur Organisasi

Sumber : LKP Multi Logika Binjai

2) Tugas dan Tanggung Jawab masing-masing Bagian

Setiap bagian atau jabatan dalam organisasi memiliki tugas dan tanggung jawab yang harus dilaksanakan, secara garis besar tugas dan tanggung jawab dari setiap bagian pada LKP Multi Logika Binjai adalah sebagai berikut:

a. Pimpinan

1) Pimpinan Lembaga mempunyai tugas memimpin dan mengawasi, mengkoordinasi pelaksanaan pelatihan kepada setiap peserta didik.

2) Menyusun rencana kerja tahunan berdasarkan analisa situasi dari hasil evaluasi pelaksanaan kegiatan tahun lalu.
3) Menyusun rencana kerja staf lima tahun bersama staf dan instansi terkait.

b. Sekretaris

1) Memberi perintah atau instruksi kepada bawahan secara resmi, baik secara lisan maupun tertulis.
2) Mengadakan rapat atau pertemuan secara bersama-sama pada suatu waktu tertentu dengan pegawai bawahan.
3) Mengadakan pengawasan secara langsung pada saat-saat tertentu kepada pegawai bawahan yang sedang melaksanakan tugasnya, yaitu pengawasan yang bersifat positif. Bila terjadi kesalahan diberi petunjuk dan pmbinaan.

c. Bendahara

1) Menerima dan membukukan keuangan
2) Menyalurkan dana sesuai dengan kebutuhan
3) Mengkonsultasikan pengeluaran dana kepada penyelenggara
4) Mengarsip tanda bukti keluar masuk uang
5) Mengamankan uang kas lembaga

d. Kepala Instruktur

1) Menyusun rencana strategis pengembangan program pelatihan
2) Menyusun rencana kerja dan anggaran tahunan program pelatihan
3) Merencanakan, melaksanakan dan mengontrol sistem pelatihan
4) Merencanakan, melaksanakan dan mengontrol pelaksanaan kurikulum pelatihan
5) Merencanakan, melaksanakan program pengembangan dan peningkatan kualitas sumber daya manusia di lingkungan pelatihan.
6) Melaksanakan prosedur penjaminan tercapainya standar mutu lulusan pelatihan.
7) Menyediakan dokumen dan pedoman pelaksanaan kurikulum pelatihan.

e. Instruktur

1) Membantu membersihkan dan memper-siapkan ruang kursus sebelum dan sesudah kursus selesai.
2) Mempersiapkan diri secara fisik dan mental.
3) Mempersiapkan bahan ajar sesuai kurikulum.
4) Melaksanakan program pengajaran dan menggunakan metode yang relevan.
5) Mengadakan evaluasi / penilaian.
6) Mengisi daftar hadir siswa.
7) Melaporkan pencapaian target kurikulum.

8) Membuat catatan-catatan khusus bagi peserta yang perlu mendapat perhatian.
9) Mempunyai target peningkatan mutu siswa.
10) Membimbing peserta kursus dengan aktif.
11) Merencanakan soal-soal/latihan dan modul bagi peserta.
12) Merencanakan calon peserta ujian.
13) Memberikan laporan berkala kepada pimpinan bagi peserta yang kurang aktif dan bagi peserta yang berprestasi.

f. Kepala Teknisi

1) Bersama Kepala Instruktur merencanakan program pengembangan lab.
2) Bertanggung jawab akan keamanan dan tata tertib di dalam lab.
3) Bertanggung jawab dalam pengelolaan administrasi dan inventarisasi kekayaan lembaga.
4) Bertanggung jawab dalam mendayagunakan sarana dan prasarana.
5) Bersama instruktur mengatur pengadaan bahan-bahan pengajaran.
6) Mengkoordinasikan keterlibatan peserta didik, instruktur dan teknisi dalam pemeliharaan dan keindahan lembaga.

g. Teknisi

1) Menampung dan menyimpan hasil praktek.
2) Menerima dan mendistribusikan alat dan bahan praktek.
3) Mengadministrasikan alat dan bahan praktek.

4) Memeriksa keadaan peralatan pelatihan.
5) Menerima informasi kerusakan pada peralatan pelatihan dari pemakai dan memperbaiki terjadinya kerusakan dan cara pelaksanaan perbaikannya.
6) Memelihara kebersihan ruangan dan penyimpanan alat-alat yang telah dan akan diperbaiki supaya teratur rapi.

h. Front Office

1) Memastikan meja depan dalam keadaan rapi dan memiliki semua alat tulis dan materi yang diperlukan (misalnya pena, formulir dan selebaran informatif).
2) Melatih, mengawasi dan mendukung staf lembaga.
3) Menangani keluhan dan permintaan.
4) Memecahkan masalah keadaan darurat.
5) Memantau stok dan pesan perlengkapan lembaga.
6) Memastikan distribusi surat yang benar.
7) Menyiapkan dan pantau anggaran lembaga.
8) Menyimpan catatan biaya dan biaya lembaga yang diperbarui.
9) Memastikan kebijakan dan persyaratan keamanan lembaga terpenuhi.

2.15 Adapun perbedaan penelitian sebelumnya

Tabel 2.6 Perbedaan penelitian lainnya dapat dilihat pada tabel 2.6 Berikut :

No	Nama Peneliti	Judul	Hasil Peneliti
1.	Alex sandro irawan 2018	Pengembangan *Intrusion Detection System* Terhadap SQL *Injection* Menggunakan Metode *Learning Vector Quantization*	Evaluasi tingkat akurasi dilakukan dengan cara menguji aplikasi dengan menggunakan data query yang bervariasi kedalam algoritma *learning vector quantization* ketika aplikasi terpasang pada suatu jaringan. Dengan menggunakan *parameter* dengan hasil akurasi yang paling maksimal didapatkan akurasi pada aplikasi deteksi SQL *injection* mencapai 80%.
2.	Laksono Adiputro AR 2018	*Portable Web Penetration Test Tool Memanfaatkan Single Board PC*	Berhasil menguji aplikasi penetrasi testing yang telah dirancang dengan cara membandingkan aplikasi sejenis. Hasil dari perbandingan tersebut yaitu aplikasi Nenggala menjalankan proses mengeluarkan database lebih cepat dengan waktu 55 detik.
3.	Sutarti 2017	PERANCANGAN DAN ANALISISKEAMANA NJARINGAN	Sistem *honeypot* telah berhasil meringankan tugas dari deteksi menjadi lebih

		NIRKABEL DARI SERANGAN DDOS (*DISTRIBUTED DENIAL OF SERVICE*) BERBASIS *HONEYPOT*	sederhana,efektif dan murah. Konsepnya sendiri sangat mudah dipahami dan diimplementasikan.*Honeypot* sendiri ditujukan untuk mendeteksi serangan yang dilakukan oleh *attacker* dengan mengecoh *attacker* tersebut dengan fasilitas *mirror server*.
4.	Agus Tedyyana, 2016	MEMBUAT *WEB SERVER* MENGGUNAKAN *DINAMIC DOMAIN NAME SYSTEM* PADA IP DINAMIS	Penelitian berhasil dilakukan dan webserver sendiri dengan custom domain berhasil dibuat, sehingga tidak harus sewa hosting yang harganya cukup mahal. Hanya saja kelemahan menggunakan webserver sendiri yaitu apabila komputer kita mati, internet speedy gangguan, maka website tidak akan bisa di akses.
5.	Muhammad Yasin Simargolang 2017	IMPLEMENTASI KRIPTOGRAFI RSA DENGAN PHP	Dari hasil pengujian respon sistem enkripsi dekripsi dihasilkan waktu akses yang cenderung lebih kecil, apabila pada saat proses enkripsi berlangsung menggunakan bilangan prima dan kunci.

6.	Firlhi Kurniawan 2017	Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost.	Analisis pengujian waktu pemrosesan pada saat *login*, SHA-1 lebih unggul daripada SHA-3.Tetapi perlu diingat, hal ini juga tidak bisa membuat SHA-1 lebih baik dari SHA-3, dikarenakan waktu yang ditempuh yaitu dalam *milliseconds*,
7.	Edy Budi Harjono 2016	Analisa Dan Implementasi Dalam Membangun Sistem Operasi*Linux* Menggunakan Metode LSF Dan REMASTER	Telah dilakukan percobaan desain dan mengembangkan sistem lokal melalui LFS dan mengambil beberapa langkah menuju tujuan besar dengan menjaga keuntungan Remastering.
8.	Yulianingsih 2017	Menangkal Serangan *SQL Injection* dengan *Parameterized Query*	*SQL Interpreter* tidak dapat membedakan antara perintah yang dimaksud dengan kode yang di-*inject* oleh penyerang yang kemudian dieksekusi dan mengakibatkan terekspos nya database.
9.	Ade Hendini 2016	PEMODELAN UML SISTEM INFORMASI MONITORING PENJUALAN DAN STOK BARANG (STUDI KASUS:	Dengan adanya sistem informasimonitoring penjualan dan stok barang ini, mempermudah pelaku usaha dalam memantau atau mengetahui penjualan dan

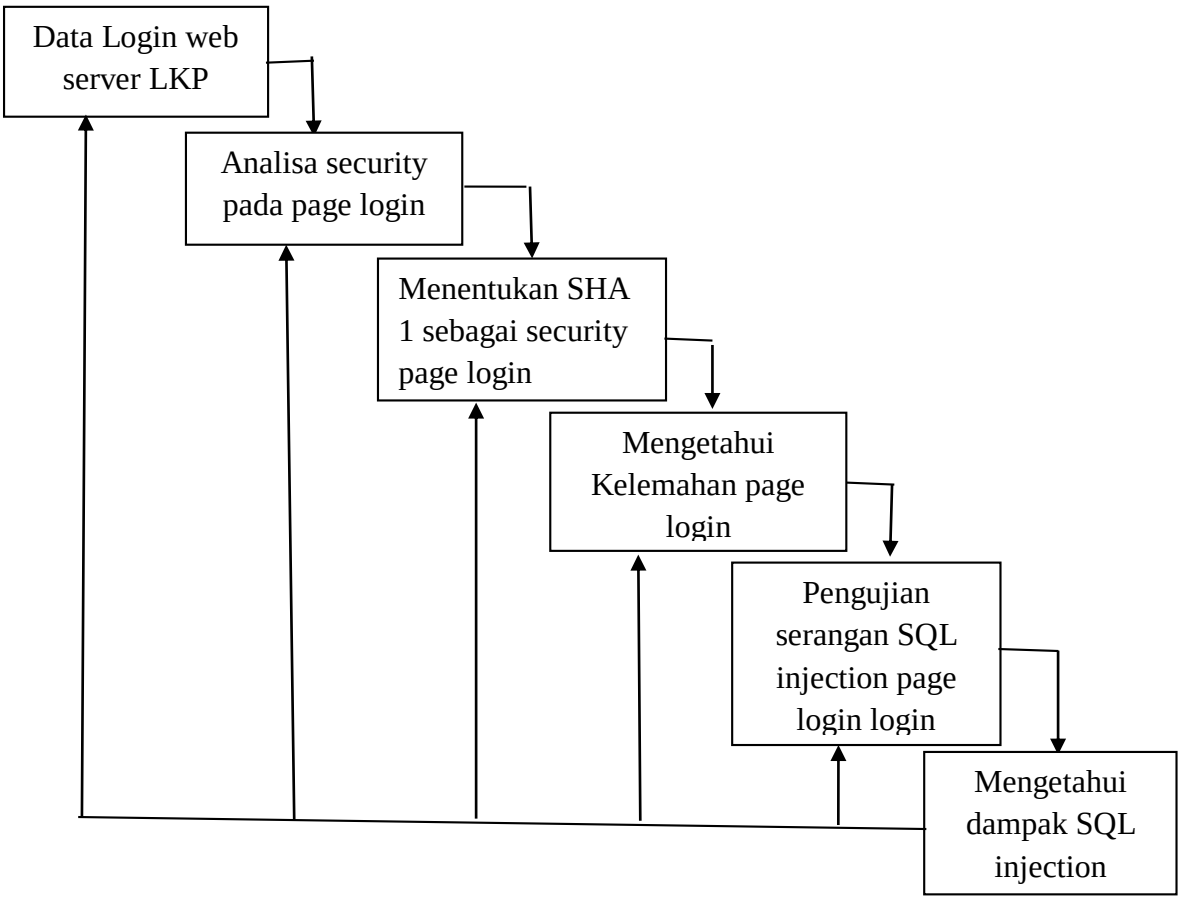
		DISTRO ZHEZHA PONTIANAK)	stok barang ditiap cabang.
10	Muhammad Suyuti Ma'sum	Analisis Perbandingan Sistem Keamanan JaringanMenggunakan Snort dan *Netfilter*	Berdasarkan pengamatan pada pengujian sistem snort dan *netfitler* dapat diketahui bahwa penggunaan *memory* pada *Server* Snort lebih sedikit dari pada *Server Netfilter*. Oleh karena itu snort lebih baik digunakan dalam mengoptimalkan *memory* pada *server*. Penggunaan *memory* sistem keamanan jaringan snort dan *netfilter*

BAB III

METODE PENELITIAN

3.1 Tahap Penelitian

Metode yang digunakan dalam membangun penelitian ini adalah dengan Metode pengumpulan data, pengumpulan data adalah tahapan yang penting dalam berlangsungnya sebuah riset, karena demi mendapatkan data yang tepat maka riset akan berlangsung sesuai dengan rumusan masalah yang telah ditentukan.



Gambar 3.1 Tahapan Penelitian

Adapun penjelasan dari tahapan – tahapan penelitian seperti gambar 3.1 diatas adalah sebagai berikut :

a. Data Login web server LKP

   Untuk pertama kali data di dapati berasal dari data awal login, dimana data login pada LKP multi logika masih terkadang dipergunakan pengguna untuk mengakses *website* LKP multi logika.

b. Analisa security pada page login

   Melakukan pengamanan *web page login* dari serangan *SQL Injection* .

c. Menentukan SHA 1 Sebagai *security page login*

   Membuat suatu pengamanan dengan menggunakan enkripsi SHA 1 dari suatu serangan *SQL Injection.*

d. Mengetahui Kelemahan *page login*

   Web page login yang ada pada LKP multi logika menggunakan penyimpanan database untuk akun pengguna *login page*, dimana web page tersebut masih menggunakan kata sandi yang belum memililki enkripsi.

e. Pengujian serangan *SQL Injection page login*

   Dengan begitu peneliti mencoba merubah sedikit perintah pemrograman yang mengarah pada kata sandi pengguna itu sendiri, kemudian menganalisa apakah kata sandi yang di enkripsi sha 1 mampu mengamankan web *page login* dari serangan *Sql Injection*.

f. Mengetahui dampak *SQL Injection.*

Banyak nya dampak yang terjadi karena adanya serangan *SQL Injection* yang dapat masuk ke dalam suatu sistem dengan melakukan pencurian informasi sensitive yang tersimpan di dalam basis data. Dengan ini melakukan *login page web server* dengan enkripsi SHA 1.

3.2 Metode Pengumpulan Data

a. Studi Pustaka

Studi Pustaka adalah tahap yang mana dengan mempelajari teori-teori yang ada dan berkaitan mengenai penelitian dalam mendukung pemecahan masalah penelitian. Dalam pencarian referensi didapati pada perpustakaan Universitas Pembangunan Panca Budi, jurnal yang berkaitan tentang *security* keamanan pada jaringan.

b. Studi Lapangan

Melakukan pencarian informasi mengenai *web server* yang ada di LKP Multi logika Binjai . Dengan mencari tahu proses keamanan *page login* portal *web server* yang ada di LKP Multi Logika, melihat topologi yang terdapat pada LKP Multi Logika mendapatkan hal informasi mengenai gangguan yang terjadi pada *web server*.

c. Wawancara

Dalam tahap ini penulis melakukan pertemuan dan wawancara kepada pihak-pihak yang berhubungan dan berpengalaman dengan penelitian ini

d. Observasi

Tahap ini bertujuan untuk memperoleh informasi mengenai perangkat dan data yang dibutuhkan yang berkaitan dengan penelitian ini.

3.3 Analisis Sistem Sedang Berjalan

Teknologi yang berkembang memiliki kemampuan untuk mengolah informasi di dalam dunia digital. Informasi yang diolah dapat bersifat rahasia sehingga dibutuhkan tempat penyimpanan yang disebut basis data. Basis data akan memanipulasi data yang disimpan dengan perintah-perintah *structured query language* (SQL). Dimana data login pada LKP Multi Logika masih terkadang dipergunakan pengguna untuk mengakses website LKP Multi Logika dan diterapkan tidak adanya keamanan server pada login page web di LKP Multi Logika. Ancaman serangan *SQL Injection* terhadap aplikasi yang menggunakan perintah *SQL* untuk memanipulasi data masih sangat banyak. Keamanan merupakan salah satu faktor penting yang harus diperhatikan dalam membangun sebuah *website*. Hal tersebut menjadi sebuah tantangan tersendiri bagi para pengembang *website*, karena tidak ada jaminan yang pasti akan defenisi aman itu sendiri. "tidak ada sistem yang benar-benar aman", bukanlah sebuah pernyataan semata, namun telah dirasakan dalam realitas. *Web server page login* merupakan *website* yang digunakan sebagai media dan sarana informasi komunikasi. Mengingat *website page login* dapat diakses secara bebas, maka dinilai perlu memperhatikan keamanan *website* dalam berhubungan dengan lingkungan luar. Terdapat beberapa cara yang dapat digunakan untuk melakukan pengujian terhadap kemanan *web server*. Salah satunya adalah dengan melakukan serangan

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*.

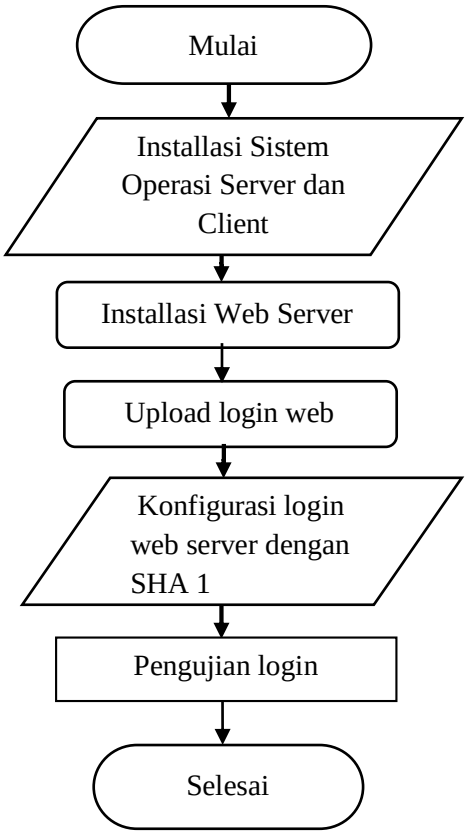
3.3.1 Sistem Yang di usulkan

Perancangan sistem adalah tahapan analisis dari siklus pengembangan sistem yang didefinisikan dari kebutuhan – kebutuhan fungsional dan persiapan untuk merancang implementasi yang menggambarkan bagaimana suatu sistem dibentuk dapat berupa penggambaran, perancangan serta pembuatan sketsa atau pengaturan, adapun beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan termasuk menyangkut konfigurasi dari komponen – komponen perangkat keras dan perangkat lunak suatu sistem sebagai berikut :

1. Berfungsi untuk meningkatkan keamanan server pada Lkp Multi Logika binjai.
2. Merancang dan menggambarkan pembuatan sketsa dengan keterangan Komponen – Komponen perangkat keras sebagai keamanan server LKP Multi Logika lebih terjamin.

3.3.2 Perancangan sistem

Dalam tugas akhir ini akan di bangun sebuah keamanan server pada login page web server dengan enkripsi SHA 1 dari serangan SQL Injection di LKP Multi Logika Binjai. Dibutuhkannya proses yang akan dibuat dalam bentuk diagram alir sebagai berikut :



Gambar 3.2 Flowchart langkah penerapan *server login page web server*

Untuk penjelasan pada gambar diatas sebagai berikut :

1. Pada sumber flowchart diatas installasi sistem operasi pada *server* dan *client*, dimana untuk *server* menggunakan sistem operasi ubuntu 18 dengan *web server apache* dan untuk client berperan sebagai *attacker* menggunakan sistem operasi kali linux.
2. Kemudian dalam membangun *web page login* dibutuhkan beberapa paket didalam *server* seperti *apache2*, *mysql-server*, bind9, dan phpmyadmin. Dimana nantinya fungsi tiap paket akan berperan masing-masing.

3. Setelah dibangunnya *web server* maka perlu adanya data atau file *web page login* itu sendiri, tentu perlu adanya data php agar *web server* dapat menyediakan *web page login*. Cara mudahnya file php yang telah di isi kode program di upload kedalam penyimpanan *web server*.
4. Sebelumnya file php yang di upload dengan perintah pemrograman didalamnya belum adanya keamanan sha1 terhadap *password user* maka perlu adanya penyesuain agar *password user web page login* dapat di enkripsi kedalam bentuk sha. Kemudian terlihat perbedaan antara *web page login* dengan perintah pemrograman tanpa enkripsi *password* sha1 dengan *password* yang di enkripsi sha1. Bahwa *web page login* dengan *password user* tanpa enkripsi sha1 tentu hanya mengikuti *password* yang di input bukan berupa kode enkripsi dan *web page login* dengan penyesuaian perintah pemrograman untuk user yang di enkripsi tentu pada kolom *password user* akan berupa kode yang telah di enkripsi sha1.

3.4 Anggaran Biaya

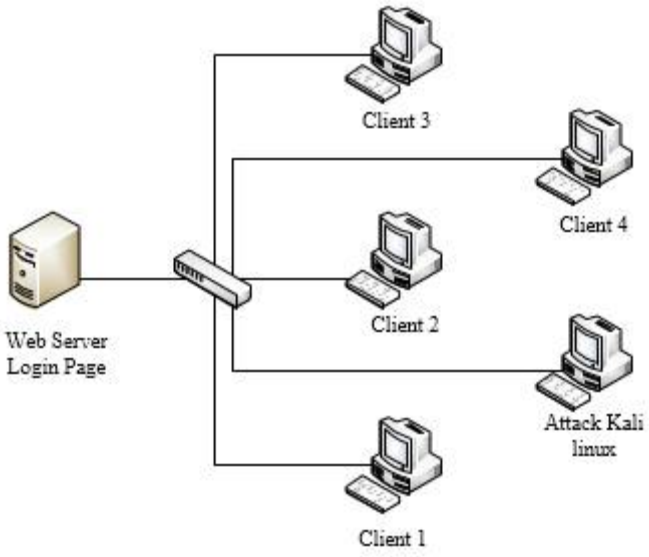
Untuk memenuhi dalam penelitian ini penulis melakukan pengumpulan biaya yang dikeluarkan untuk penelitian mengenai analisis keamanan server pada *login page web server* dengan enkripsi Sha1 dari serangan *SQL Injection*, sebagai berikut:

Tabel 3.1 Biaya Keseluruhan

No.	Hardware/Software	Spesifikasi	Jumlah Unit	Harga
1	Cable UTP 1.5 Meter + 2 RJ45	Cat 5	6	Kabel Rp. 3000/Meter RJ45 Rp. 500
2	Laptop untuk Client	Server : Intel® Core™ i3 generasi ke-4 Ram 4 GB HDD 500 GB Client: Intel Celeron B877 Ram 4Gb HDD 320 Gb	6	Client 1: Rp. 5.300.000 Client 2: Rp. 3.400.000 Client 3: Rp. 3.100.000

3.4.1 Manajemen jaringan

Sistem yang akan di bangun dalam penelitian ini dapat digambarkan dengan topologi serangan seperti berikut:



Gambar 3.3 Topologi keamanan *server* pada *login page web server*

Dalam gambar 3.3 Diatas dapat dijelaskan dengan pengalamatan IP pada tabel berikut ini:

Tabel 3.1 Daftar Alamat IP Topologi lengkap

No.	Hardware/Software Network	Port Ethernet	Alamat IP / IP Address
1	Server	Eth0	Address 192.168.43.5 Netmask 255.255.255.0 Gateway 192.168.43.1
		Eth1	Address 192.168.1.1 Netmask 255.255.255.0

2	Client Terhubung Jaringan Lokal Switch	-	Address 192.168.1.2-192.168.1.254 Gateway 192.168.1.1 DNS multilogika.lkp

Dalam tabel 3.1 Pengalamatan alamat Ip dapat dejalaskan bahwa:

1. Sistem operasi yang digunakan untuk *Server* adalah ubuntu 18, dimana *server* bekerja dalam menyediakan *login page web server* dengan enkripsi password sha1.
2. Didalam server nantinya terdapat paket-paket yang mendukung dalam membangun penelitian ini yaitu sebuah *web server*, dns server, *sql server*, dan phpmyadmin.
3. Setelah kebutuhan paket yang ada maka perlunya dalam upload data web berupa file php untuk *web page login* agar nantinya *web server* dapat menyediakan *web page login* untuk client.
4. Kemudian *client* dapat berperan sebagai *attacker* dalam melakukan sebuah serangan *Sql Injection* menggunakan perangkat lunak *sqlmap* yang memang telah ada pada sistem operasi kali linux.

Terdapat beberapa langkah dalam membangun *web server*, tentu yang paling didahulukan installasi sistem operasi itu sendiri. Pada penelitian ini jenis Operasi Sistem yang digunakan pada *Server* menggunakan Linux Ubuntu 14 dan untuk komputer *client* dan laptop penyerang menggunakan kali linux. Cara installasi dan proses installasi dari linux dominan sama yaitu Melakukan penginstallan Operasi Sistem dilakukan dengan menggunakan CD/DVD atau

Flashdisk, dalam tahap penginstallan selalu di temui pemilihan Bahasa, nama perangkat, berapa ruang partisi dan membagi ukuran untuk partisi *Primer* dan *Sekunder*.

Kemudian terdapat tahap dimana bagaimana membangun *web server* itu sendiri kebutuhan dalam membangun *web server* tentu adanya paket – paket yang dibutuhkan seperti *Apache*, Mysql, PHP, dan lainnya langkah dari penerapan paket-paket itu seperti berikut :

a. Instalasi *Apache*

Apache merupakan layanan peladen *web* paling banyak digunakan dan relatif lebih mudah untuk digunakan. Untuk memasang *Apache* pada Ubuntu 18.04 kita bisa menggunakan perintah berikut ini :

```
$ sudo apt update
$ sudo apt install apache2
```

Ikuti proses seperti menjawab Y untuk memastikan pembaca yakin melakukan instalasi *Apache* dan memasukkan kata sandi agar proses dapat berjalan. Jika sudah selesai maka saat ini *Apache* telah terpasang pada komputer pembaca. Silahkan untuk mengujinya.

b. Instalasi MySQL

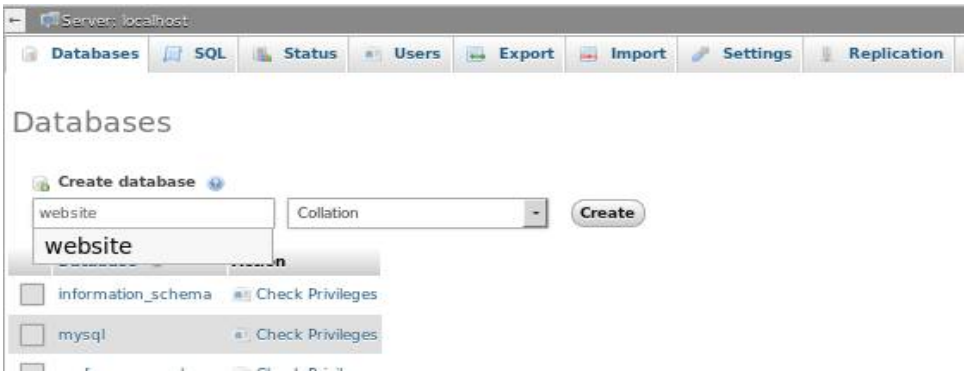
MySQL merupakan sistem manajemen basis data yang menggunakan bahasa *SQL*. Layanan ini telah digunakan lama, saya sendiri menggunakannya sejak awal belajar *server* yaitu sekitar tahun 2008. Untuk instalasi aplikasi ini bisa dengan mengetikkan perintah berikut pada terminal:

apt-get install mysql-*server* Jika ada pertanyaan Y/n silahkan pilih Y saja dan tunggu proses instalasi selesai. Saat ini pada sistem kita telah terpasang *MySQL* namun karena konfigurasi yang digunakan masih menggunakan bawaan maka nama penggunanya adalah *root* dan kata sandinya tidak usah diisi.

c. Instalasi PHP

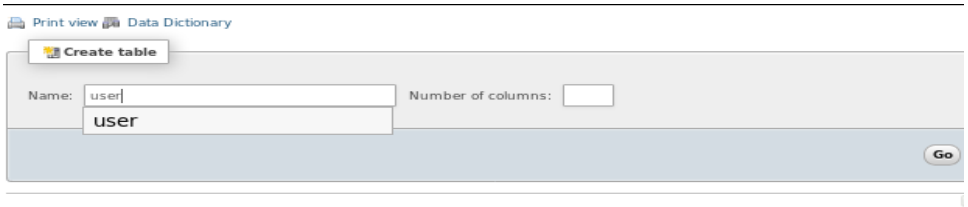
PHP adalah paket yang harus dipasang pada peladen *web* jika kita ingin menggunakan *web* berbasis php. Untuk instalasi nya silahkan mengetikkan perintah berikut pada terminal :

apt install php libapache2-mod-php php-mysql kemudian setelah diterapkannya paket-paket yang dibutuhkan maka masuk pada tahap dimana dilakukannya upload file php yang nantinya digunakan *web server* dalam menyediakan *web page login* untuk user. Akan tetapi sebelum user dapat menikmati *web login page* dalam bentuk akun untuk masuk kedalam *web page login* maka *web page login* membutuhkan penambahan database untuk *web page login* seperti berikut:



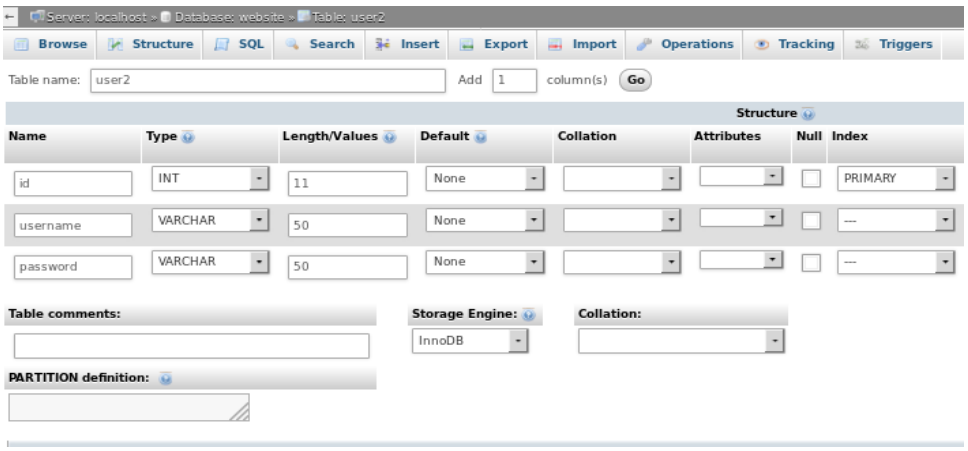
Gambar 3.4 Tampilan membuat *Database*

Dari gambar yang ada diatas dapat dijelaskan bahwa dilakukannya pembuatan Database untuk website yang akan dibangun. Dalam membangun database untuk *login page* ini peneliti menggunakan *phpmyadmin*, dikarenakan mempermudah peniliti dalam membangun database itu sendiri tanpa repot melakukannya dengan *command line* pada linux. Setelah dilakukan pembuatan database maka buatlah struktur tabel untuk database seperti berikut:



Gambar 3.5 Tampilan membuat Tabel

Sesuai yang ada pada gambar penulis menambahkan tabel *user* kedalam *database website* dengan jumlah kolom sebanyak 3 kolom yang mana nantinya kolom tersebut berisikan id, *username*, dan *password* dari pengguna *login page* seperti berikut:



Gambar 3.6 Tampilan membuat kolom Tabel

Mengisi kolom pada tabel dapat dilakukan pada saat setelah tebel dibuat, seperti pada gambar dapat dijelaskan bahwa *login page webserver* membutuhkan database dengan nama *website* kemudian tabel dengan *user* dan isi dari tebel itu sendiri id, *username*, dan *password*. Fungsi dari database ini nantinya agar *website* dapat melayani pengguna dalam menyimpan data akun pengguna kedalam *query* database yang ada pada *MySQL*.

Kemudian setelah database yang dibangun telah selesai dan siap digunakan maka masuk pada pembuatan form dari *login page webserver* dan bagaimana proses dari file PHP *login page* terhubung dengan database *MySQL* yang telah dibuat.

3.5 *Security* SHA1

Hash adalah algoritma enkripsi untuk mengubah text menjadi deretan karakter acak. Jumlah karakter hasil *hash* selalu sama. Hash termasuk *one-way encription*, membuat hasil dari hash tidak bisa dikembalikan ke text asli.

SHA1 atau *Secure Hash Algorithm* 1 merupakan salah satu algoritma *hashing* yang sering digunakan untuk enkripsi data. Hasil dari sha1 adalah data dengan lebar 20 *byte* atau 160 bit, biasa ditampilkan dalam bentuk bilangan heksadesimal 40 digit. Contoh penggunaan SHA1 untuk mengamankan *password* akun *user* seperti berikut:

```
<?php
  session_start();
  require_once("koneksi.php");
  $username = $_POST['username'];
  $pass = sha1($_POST['password']);
  $cekuser = mysql_query("SELECT * FROM user WHERE username = '$username'");
  $hasil = mysql_fetch_array($cekuser);
  if(mysql_num_rows($cekuser) == 0) {
    echo "<div align='center'>Username Belum Terdaftar! <a href='login.php'>Back</a></div>";
  } else {
    if($pass <> $hasil['password']) {
      echo "<div align='center'>Password salah! <a href='login.php'>Back</a></div>";
    } else {
      $_SESSION['username'] = $hasil['username'];
      header('location:index.php');
    }
  }

?>
```

Pada perintah yang di beri warna merah dapat dijelaskan bahwa pada saat dilakukannya registrasi untuk akun *user* maka *password* dari akun *user* tersbut akan di enkripsi kedalam algoritma *hash* atau SHA1.

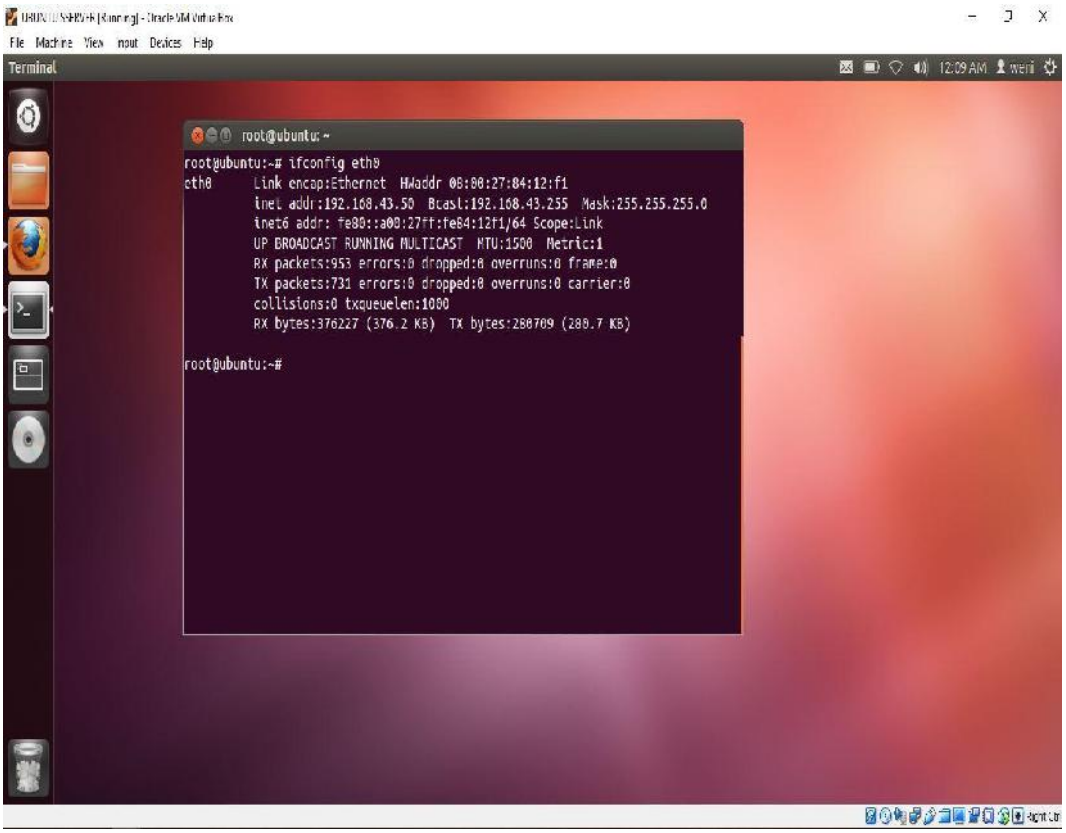
BAB IV

HASIL DAN PEMBAHASAN

4.1 Tampilan Hasil

Dalam melakukan implementasi dan evaluasi sistem *login* dengan keamanan SHA1 pada LKP Multi Logika Binjai digunakan beberapa jenis *software*, antara lain yaitu ubuntu *server*.

Linux ubuntu *server* berfungsi untuk melakukan penyimpanan data website login dan database *user*.



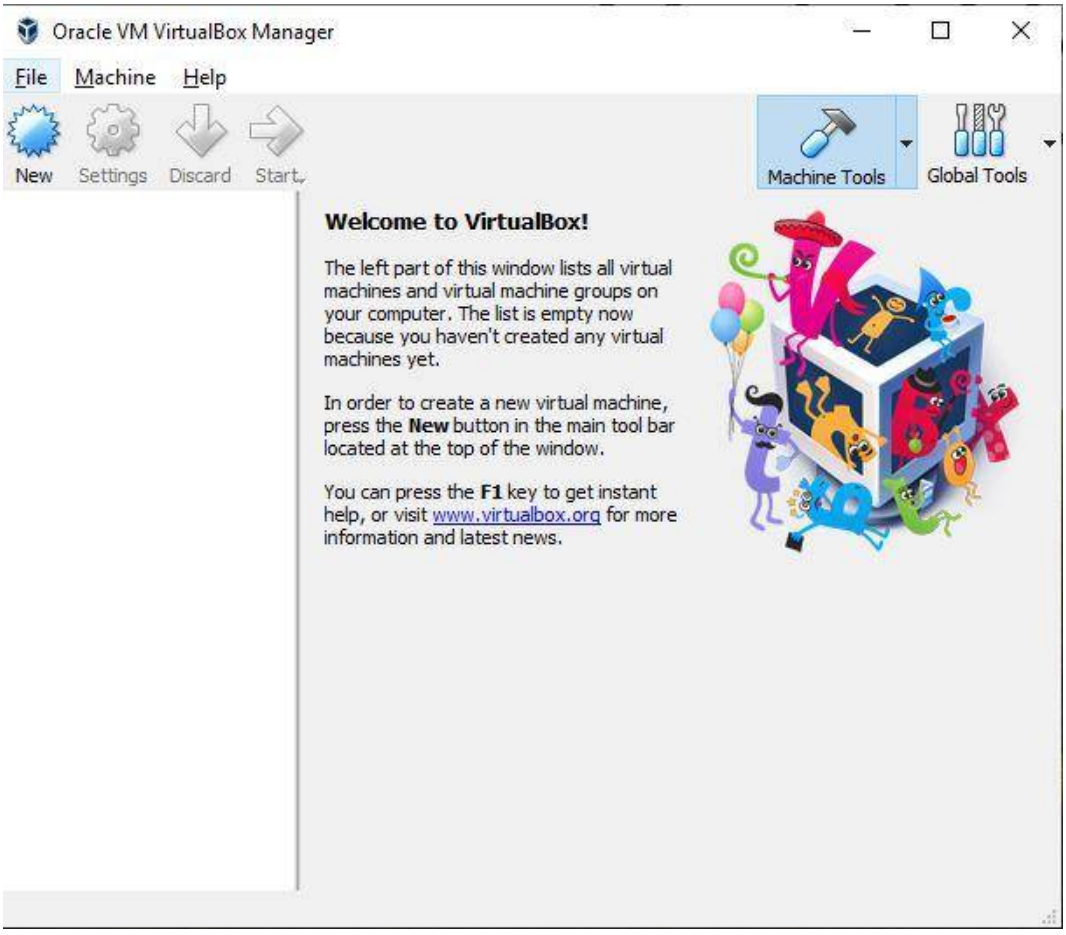
Gambar 4.1 Tampilan *Ubuntu Server*

1.2 Pengujian Aplikasi dan Pembahasan

Dalam rancangan sistem penulis merancang dan menginplementasikan sistem *login* dengan menggunakan keamanan data SHA1 dengan *VirtualBox*, dan tahapan konfigurasi yang harus dilakukan adalah sebagai berikut :

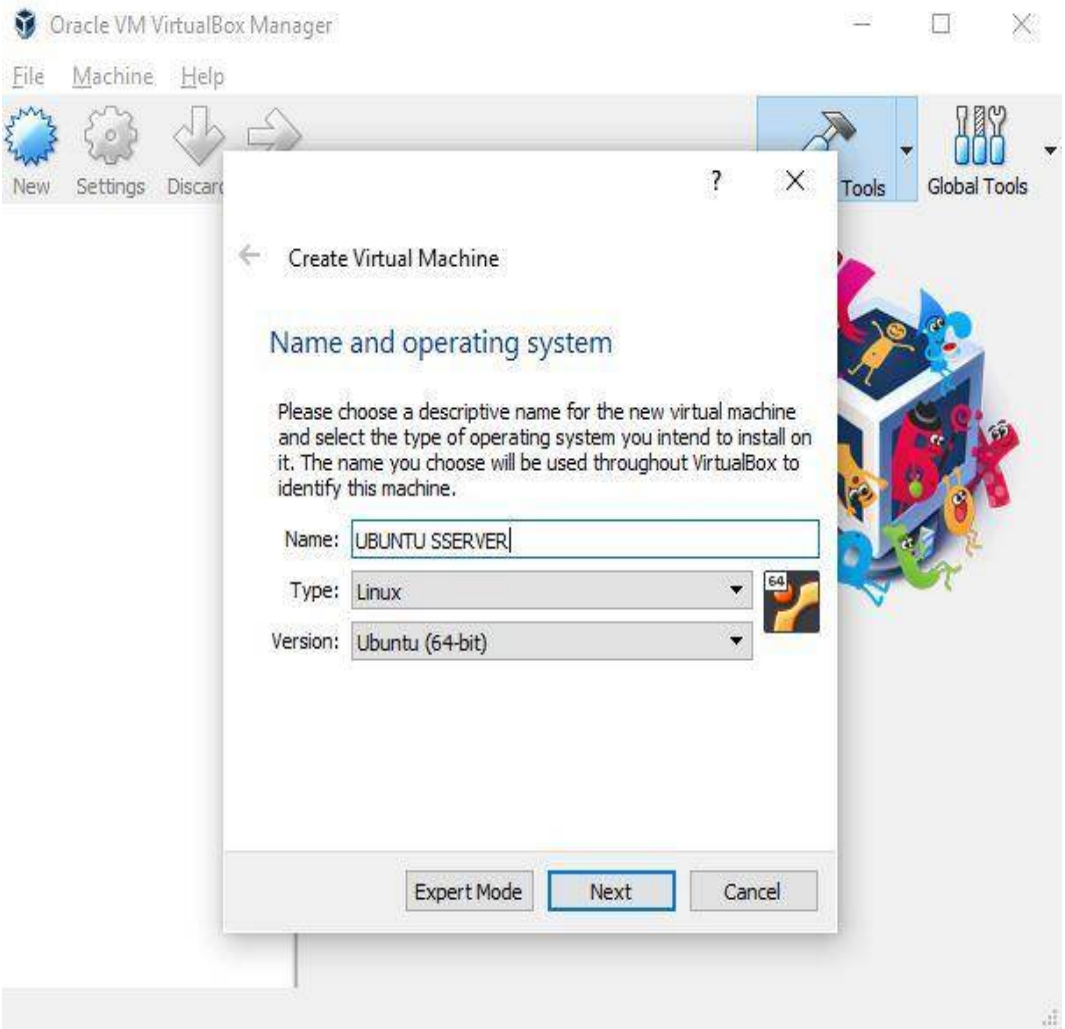
1. **Konfigurasi *VirtualBox***

   1) Langkah awal dalam melakukan instal sistem operasi *linux ubuntu server* ini adalah dengan melakukan klik pada *new virtual machine wizard.*



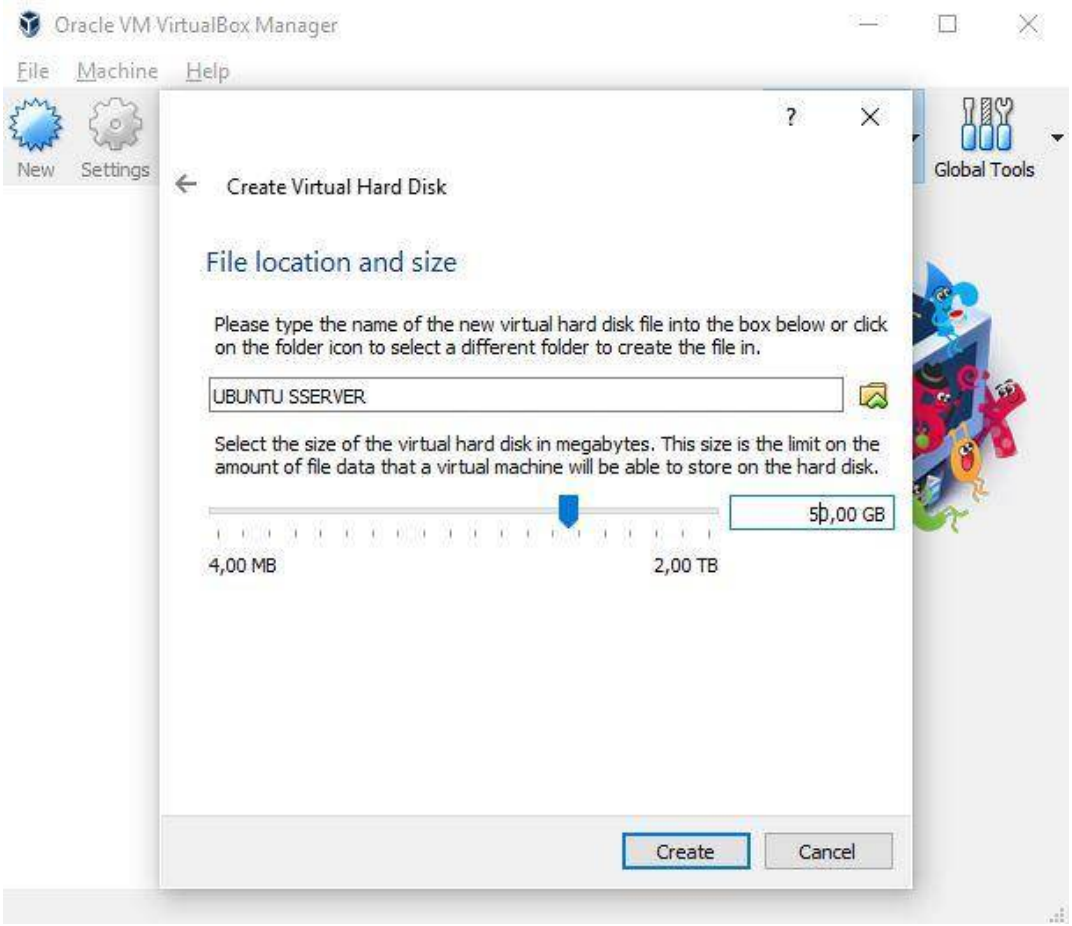
Gambar 4.2 Tampilan Awal *VirtualBox*

2) Setelah itu, klik NEW untuk membuat nama pada sistem operasi baru yang akan kita lakukan penginstalan, Dalam hal ini penulis memberikan nama dengan “UBUNTU SSERVER”.



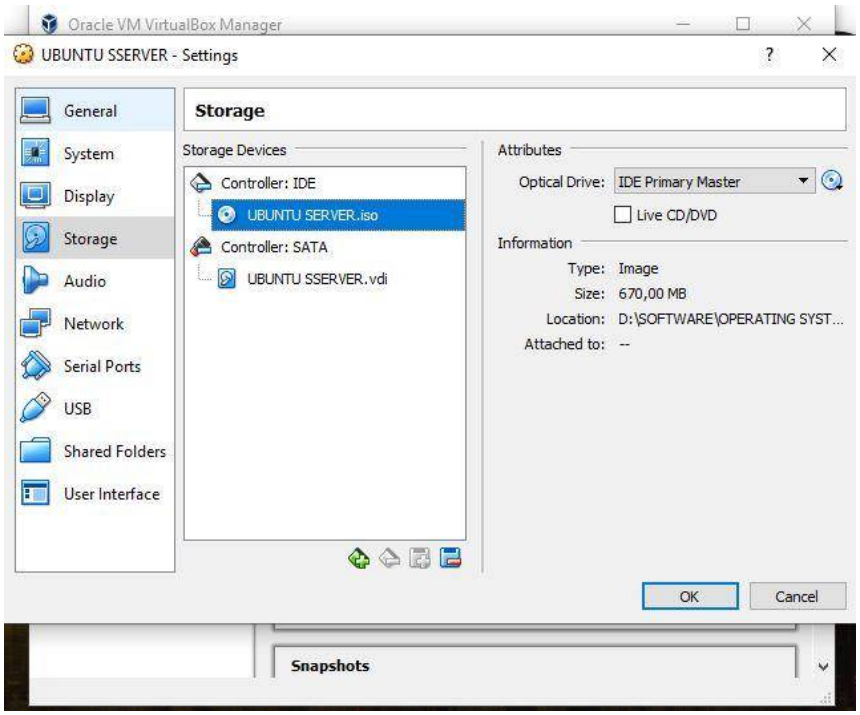
Gambar 4.3 Memberikan Nama Sistem

3) Kemudian dapat menentukan jumlah partisi yang digunakan untuk menginstal sistem operasi ubuntunya. Dalam hal ini penulis memberikan partisi 50 GB pada sistem.



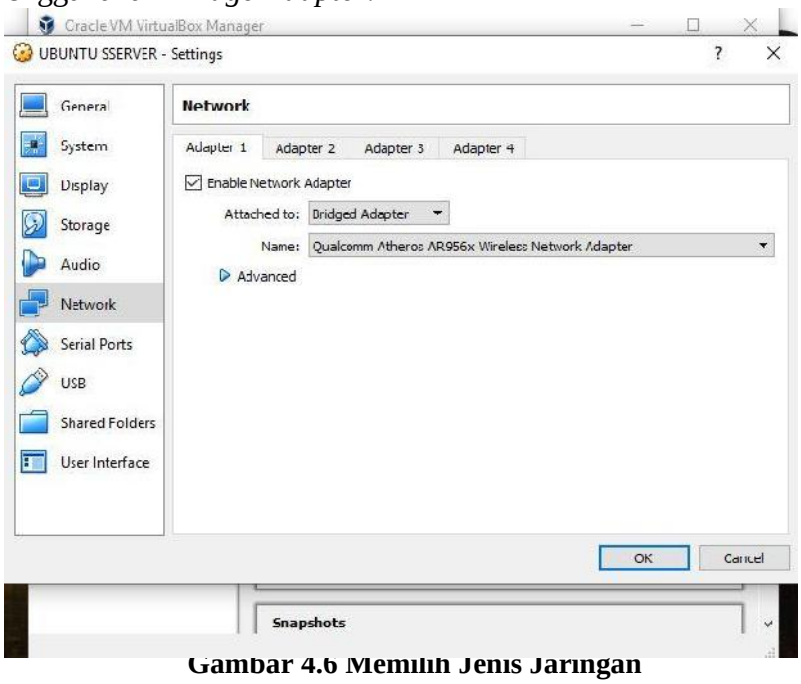
Gambar 4.4 Memberikan Partisi Harddisk

4) Kemudian menentukan file ISO sistem operasi *linux ubuntu server* tersebut. Dengan cara klik setting pada menu VirtualBox, pilih storage klik pada *Controller:IDE* kemudian, klik pada adds optical drive, ketika muncul pertanyaan pilih choose disk, lalu kita akan memilih iso yang akan kita install.



Gambar 4.5 Memilih ISO Ubuntu Server

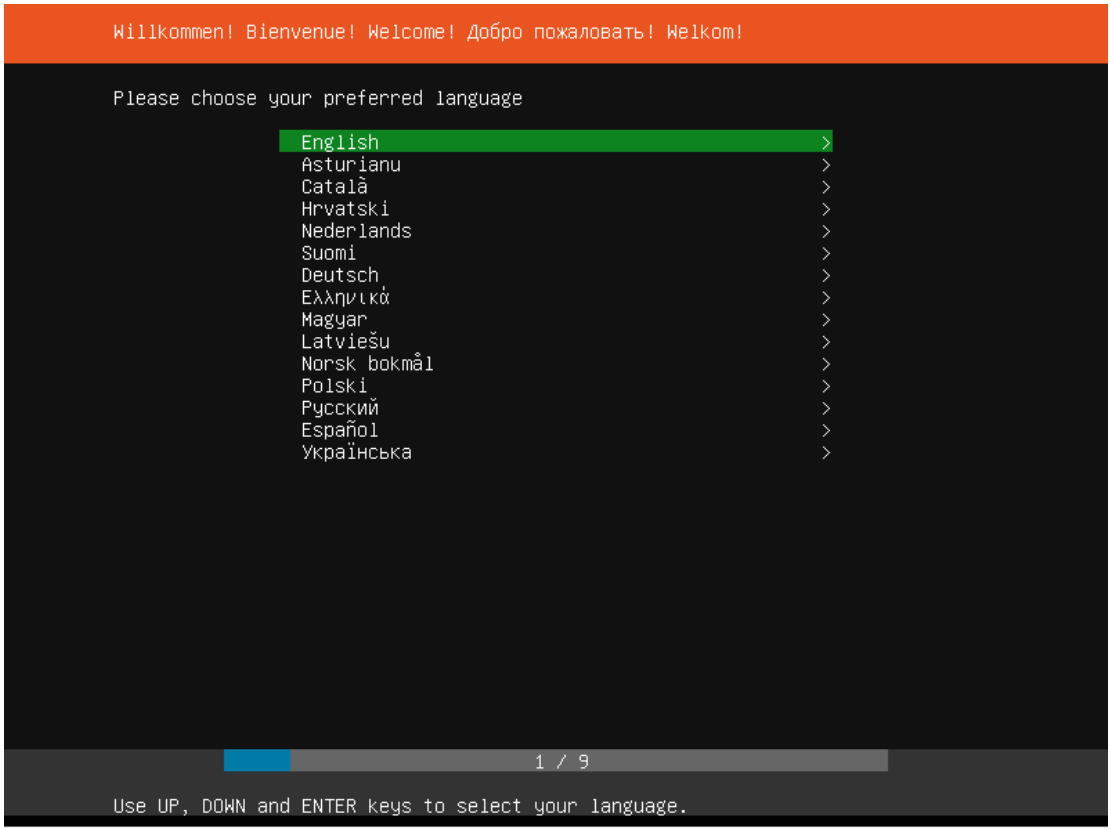
5) Lalu memilih jenis jaringan yang digunakan, disini penulis menggunakan *Bridge Adapter.*



Gambar 4.6 Memilih Jenis Jaringan

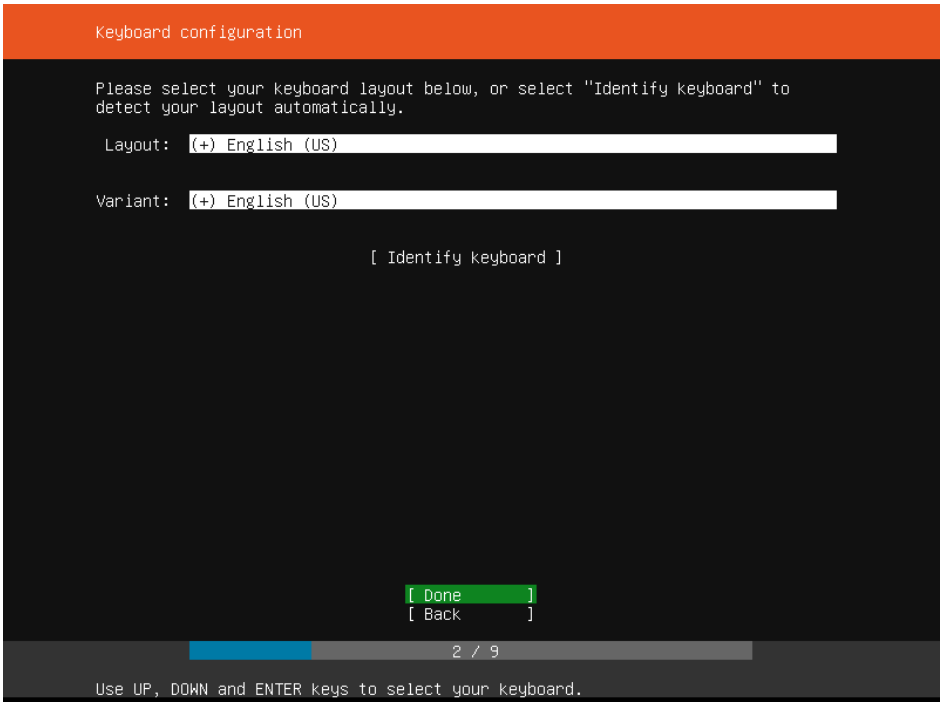
2. Instalasi Sistem Operasi Ubuntu Server

1) Langkah awal yaitu menjalankan *virtual machine* yang telah dibuat sebelumnya untuk menjalankan sistem operasi ubuntu *server*. Selanjutnya dalam tahap permulaan dipilih nama bahasa yang digunakan.



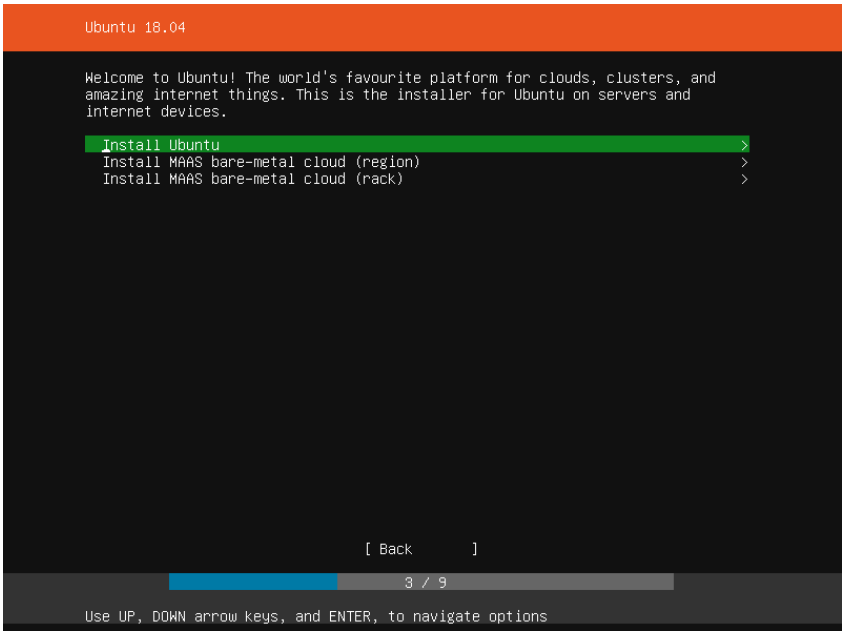
Gambar 4.7 Tampilan Menu Bahasa

2) Kemudian akan tampil tampilan pemilihan *layout keyboard* yang digunakan.



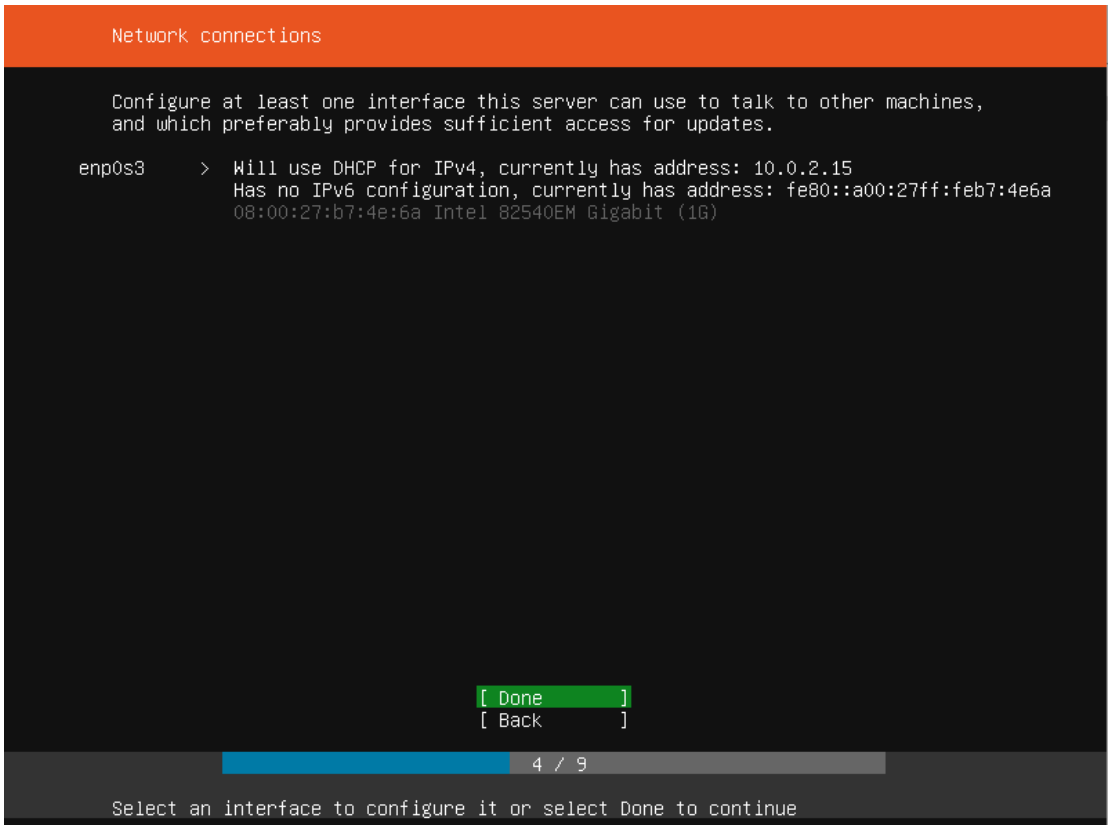
Gambar 4.8 Proses *layout keyboard*

3) Lalu akan tampilan pilihan dalam pemilihan *core* dari linux ubuntu *server*. Dalam hal ini penulis memilih menu instal ubuntu *server*.



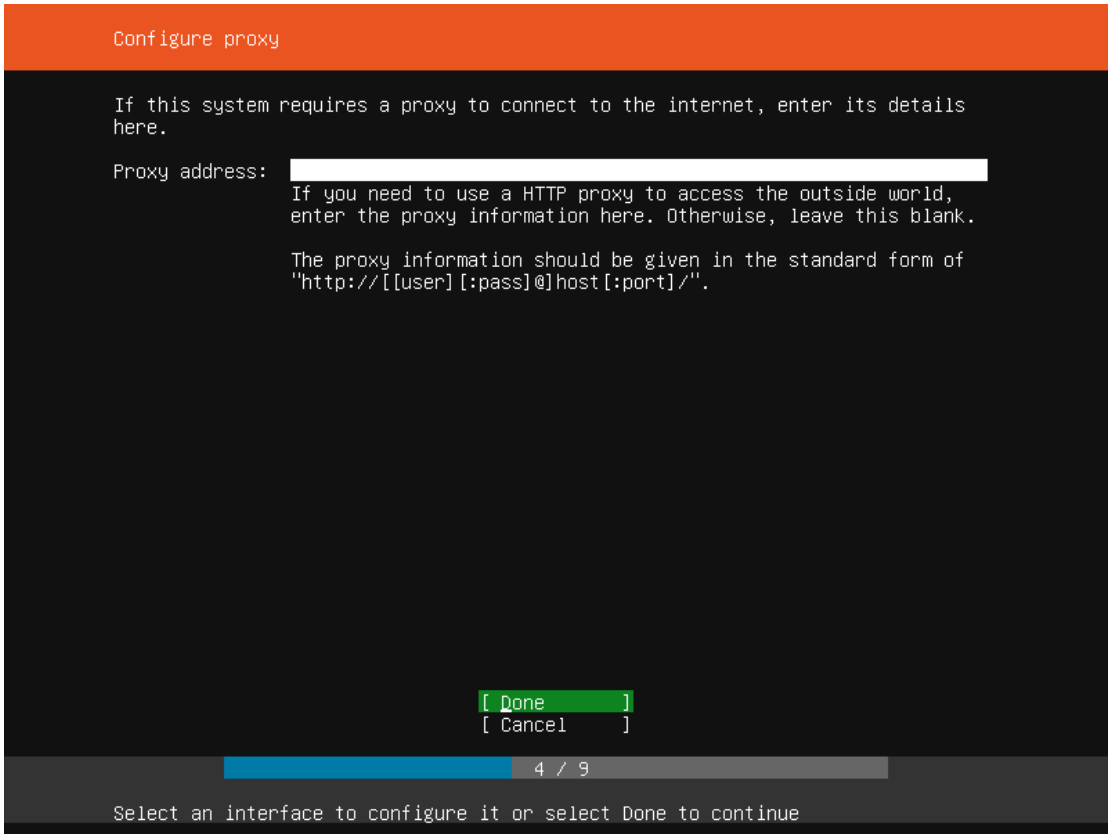
Gambar 4.9 Tampilan pilih jenis sistem

4) Setelah itu akan tampil *network card* yang terdeteksi pada sistem komputer yang akan diinstalkan ubuntu *server*.



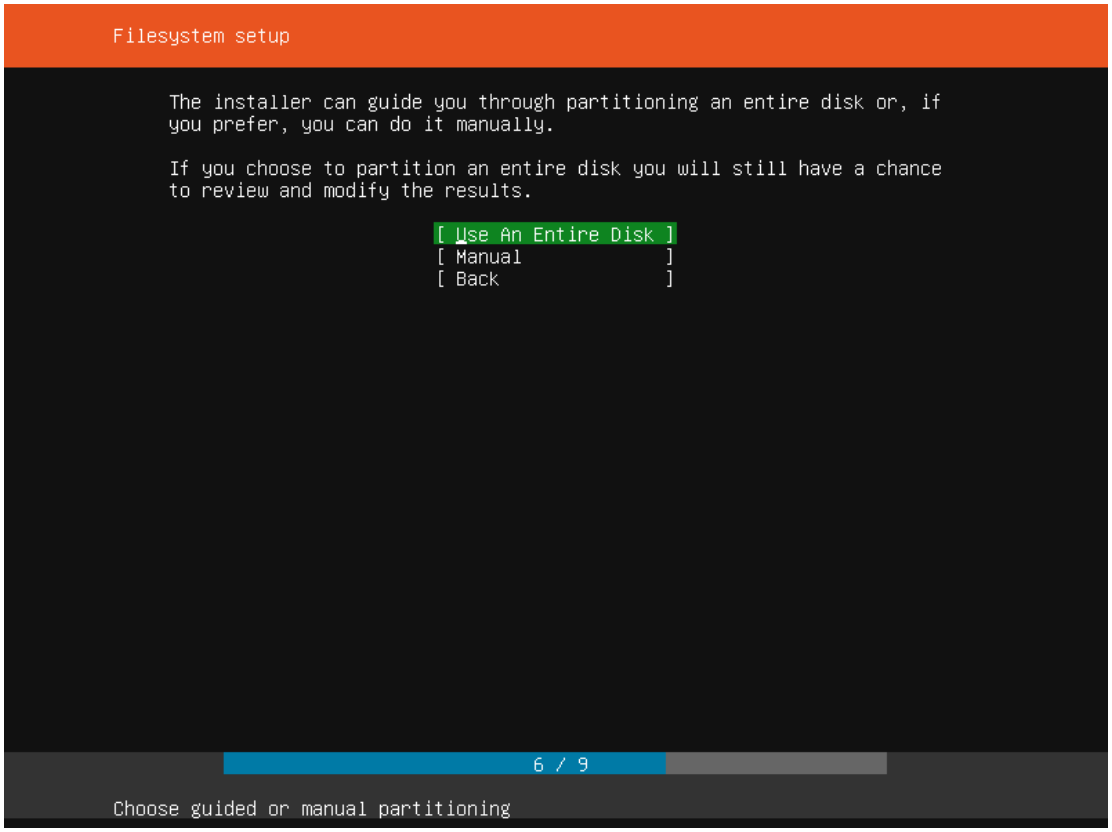
Gambar 4.10 Tampilan network

5) Kemudian akan tampil dalam *proxy* internet yang digunakan pada jaringan. Jika tidak terdapat *proxy* pada jaringan dapat langung melewati langkah ini.



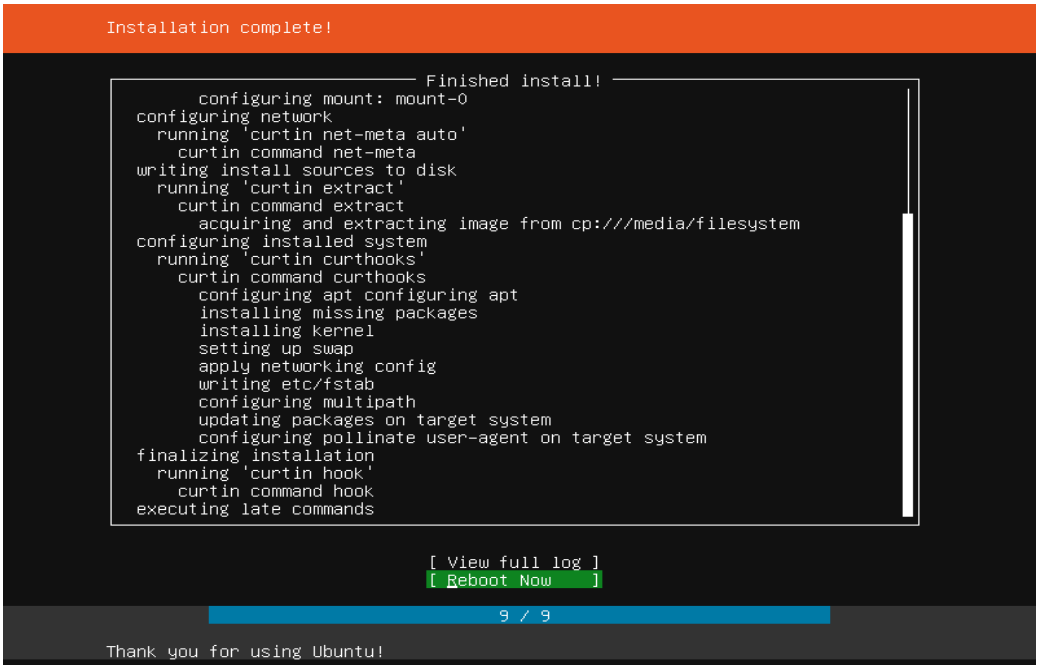
Gambar 4.11 Tampilan proxy

6) Kemudian setelah melewati tahap *proxy*, akan tampil penyimpanan harddisk yang tersedia pada sistem komputer yang akan diinstal sistem operasi linux ubuntu.



Gambar 4.12 Tampilan harddisk

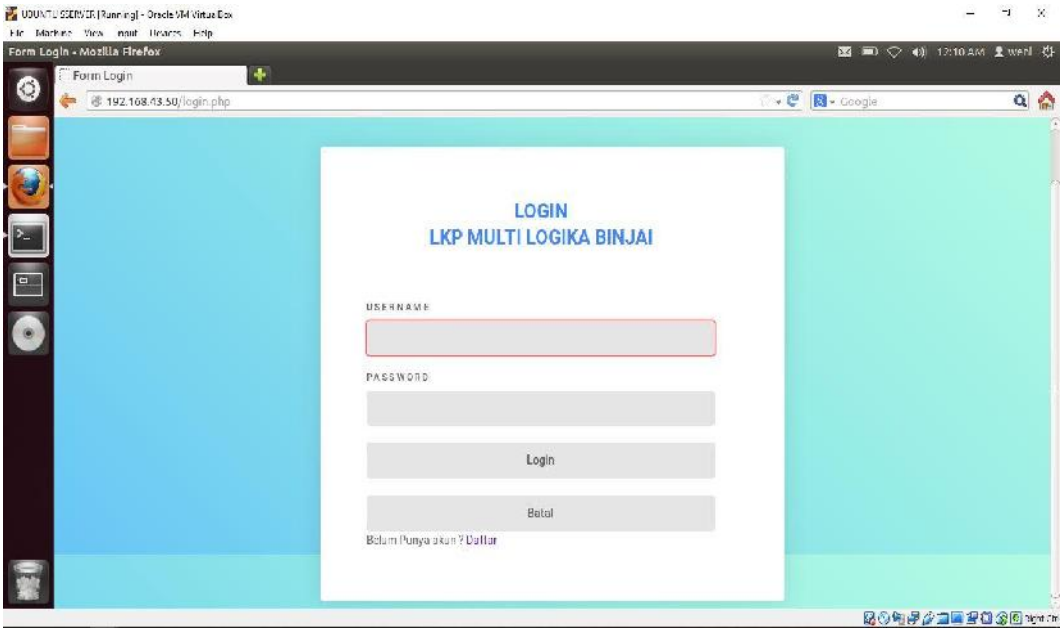
7) Setelah selesai dalam melakukan konfigurasi sistem, akan ditampilkan tampilan *user* data yang akan dijadikan sebagai *user* administrator pada sistem operasi linux ubuntu yang akan diinstal. Setelah selesai melakukan pengisian data, sistem akan melakukann penginstalan sistem operasi pada komputer.



Gambar 4.13 Tampilan instal ubuntu server

1.3 Pengujian SQL Injection Login

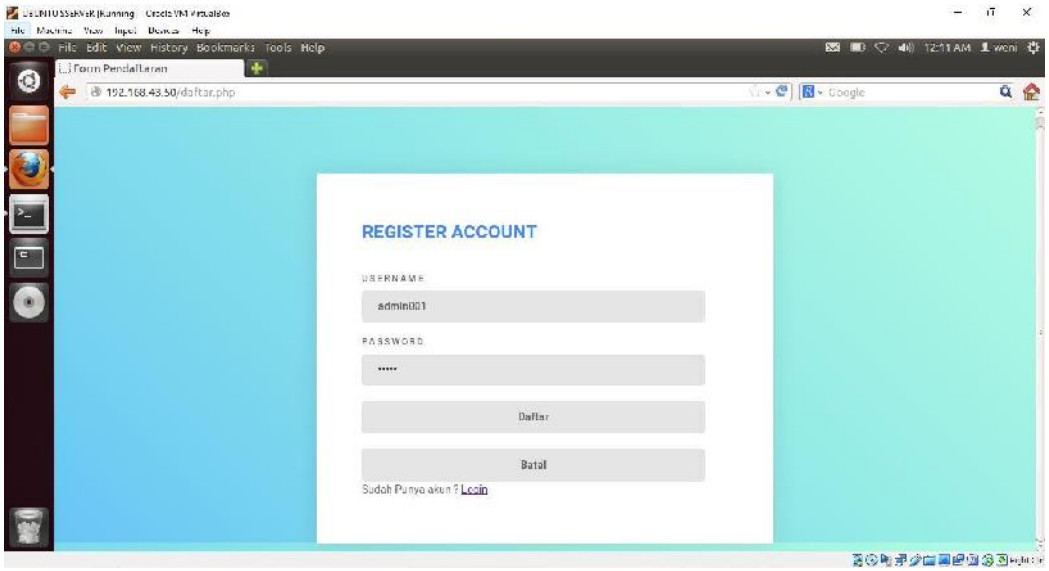
1.3.1 Tampilan Sistem Login



Gambar 4.14 Tampilan Sistem Login

Pada Gambar 4.14 Adalah tampilan login untuk masuk ke dalam website pada LKP Multi Logika Binjai

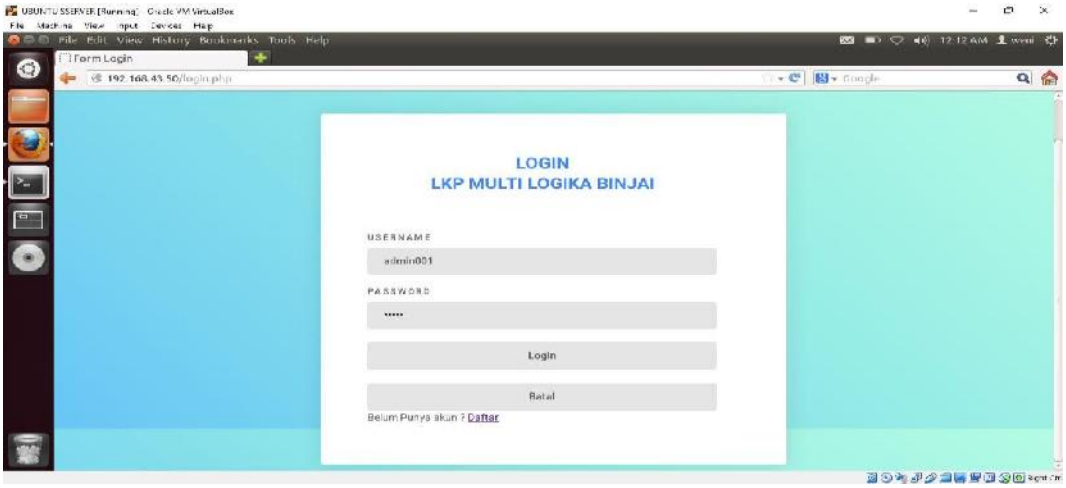
1.3.2 Melakukan *Register Account*



Gambar 4.15 Tampilan *Register Account*

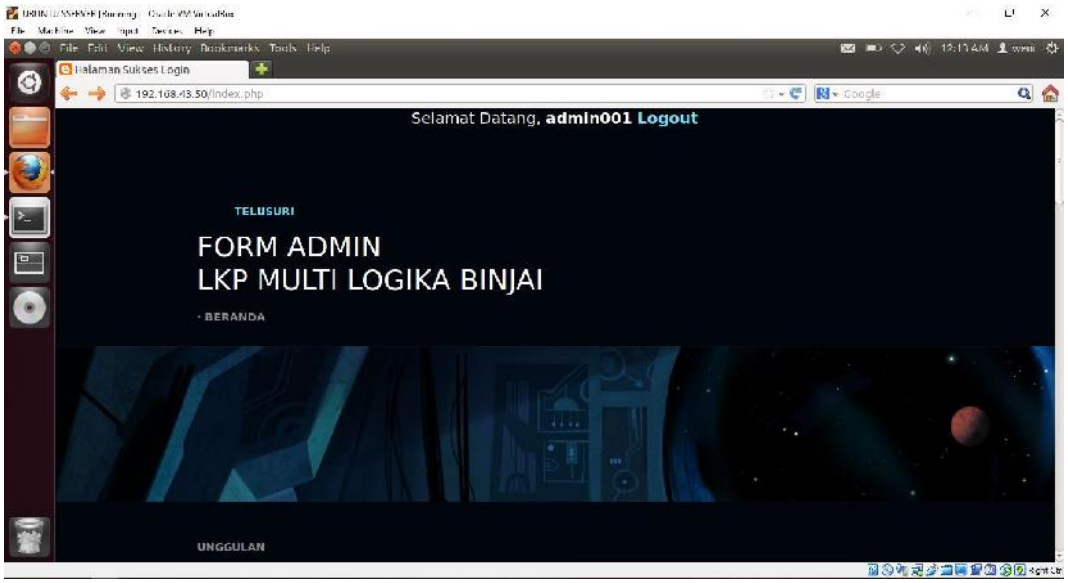
Pada Gambar 4.15 Adalah tampilan login masuk ke dalam website LKP Multi Logika binjai, yang menggunakan username dan passworrd untuk login, apabila belum terdaftar maka bisa mendaftar terlebih dahulu.

1.3.3 Melakukan Login



Gambar 4.16 Tampilan Login *Account*

Pada Gambar 4.16 adalah Tampilan dari Ubuntu Server yaitu Setelah melakukan pendaftaran, penulis mencoba login dengan akun yang sudah didaftar, setelah login sistem langsung mengarah ke website admin LKP Multi Logika Binjai, Dimana fungsi ubuntu server untuk mengenkripsi kan SHA 1 dari sistem login tersebut.



Gambar 4.17 Tampilan *Website Admin*

Pada Gambar 4.17 adalah tahap pengujian login kedalam sistem dengan menggunakan *sql injection* ini hanya dibutuhkan kode-kode sederhana yang diinputkan ke kolom username dan password yang ada ditampilan *login*. Dalam pengujian pertama ini penulis menggunakan sistem operasi ubuntu versi desktop dalam melakukan pengetesan login yang ada. Pada tahap ini, username dan password masih karakter normal yang tidak dilakukan enkripsi menggunakan SHA1 kedalam database.

1.4 Instalasi Sistem Operasi Kali Linux

1. Buka Virtualbox yang sudah di download dan instal sebelumnya

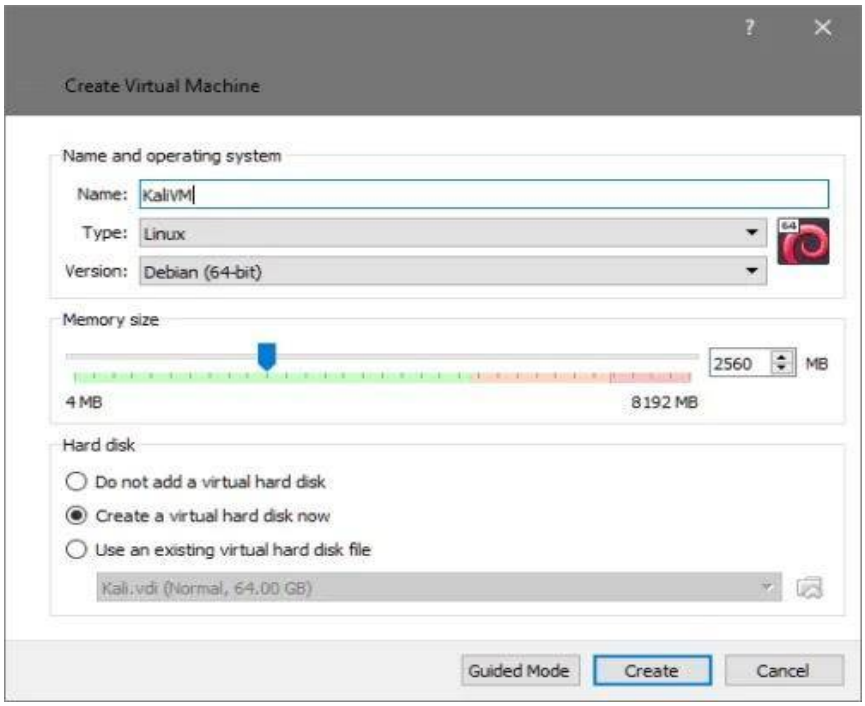
2. Pilih ‘New’ untuk menambahkan sistem operasi yang baru

3. Pilih platform dan sistem operasi yang ingin digunakan. Pilih Type ‘Linux’ dan version yang sesuai dengan OS dan platform yang kamu miliki. Klik ‘Next’

4. Atur ukuran memori yang di instal. Lalu klik ‘Next’

5. Klik radio button ‘Create a virtual hard disk now’ untuk membuat HD virtualbox lalu klik ‘Next’

6. Pilih kembali folder Kali Linux yang tadi sudah pilih kemudian klik ‘Create’



Gambar 4.18 Tampilan Instalasi awal

1. Klik *Settings*
2. Masuk ke tab *General* dan klik *Advanced*
3. Pastikan *Shared Clipboard* dan *Drag'n'Drop* adalah *Bidirectional* karena akan ada data transfer antara *host Linux* dan Kali
4. Selanjutnya masuk ke system, klik *Motherboard* dan pastikan untuk mencentang *Enable* EFI
5. Kamu bisa melanjutkan pengaturan untuk beberapa kategori tab lainnya atau membiarkannya *by default* untuk sekarang

Gambar 4.19 Tampilan *Settings*

1. Setelah aplikasi Kali Linux terbuka, pilih '*Graphic Installer*'

2. Kemudian pilih bahasa, negara, dan bahasa untuk pengaturan *keyboard*

Gambar 4.20 Tampilan Pengaturan Bahasa

3. Tentukan konfigurasi host name

4. tentukan dan masukkan *root password*

Gambar 4.21 Tampilan konfigurasi Host name

5. Lakukan konfigurasi waktu

6. Pilih *Guided - use entire disk*

1. Pilih ATA VBOX HARDISK dan klik '*Continue*'

Gambar 4.22 Tampilan Konfigurasi Waktu

2. Tentukan pilihan partisi dan klik '*Continue*'

3. Pilih *Finish partitioning* dan klik '*Continue*'

10. Pilih radio *button Yes* untuk memformat hardisk virtual dan klik '*Continue*'

11. Tunggu proses instalasi hingga selesai

12. Pilih menginstal network mirror atau tidak kemudian pilih *No*

13. Tunggu proses instalasi hingga selesai

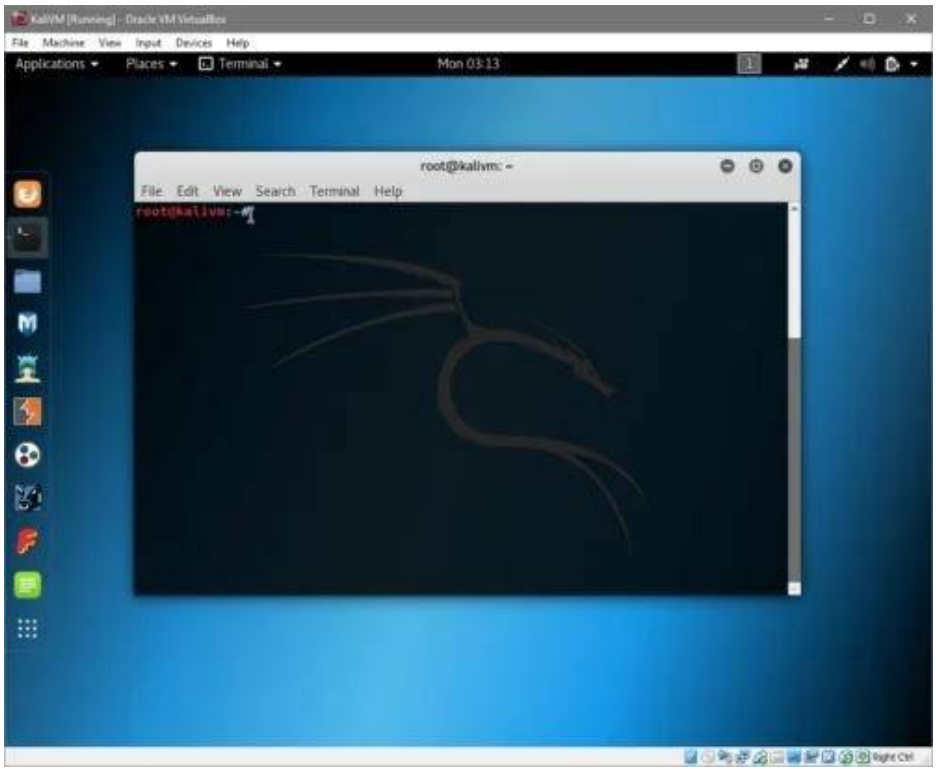
14. Kemudian pilihan untuk menginstal Grub *boot loader*, klik ‘Yes’ kemudian jangan lupa memilih tempat untuk menginstalnya setelah klik ‘*Continue*’

15. Tunggu proses instalasi hingga selesai dan klik ‘*Continue*’

16. Jika proses instalasi berhasil maka akan muncul aplikasi Kali Linux dengan menampilkan *field username*.

17. Masukkan *username* dan *password* berdasarkan dengan yang sudah di atur

18. Maka Tampilan Kali Linux di virtualbox telah berhasil

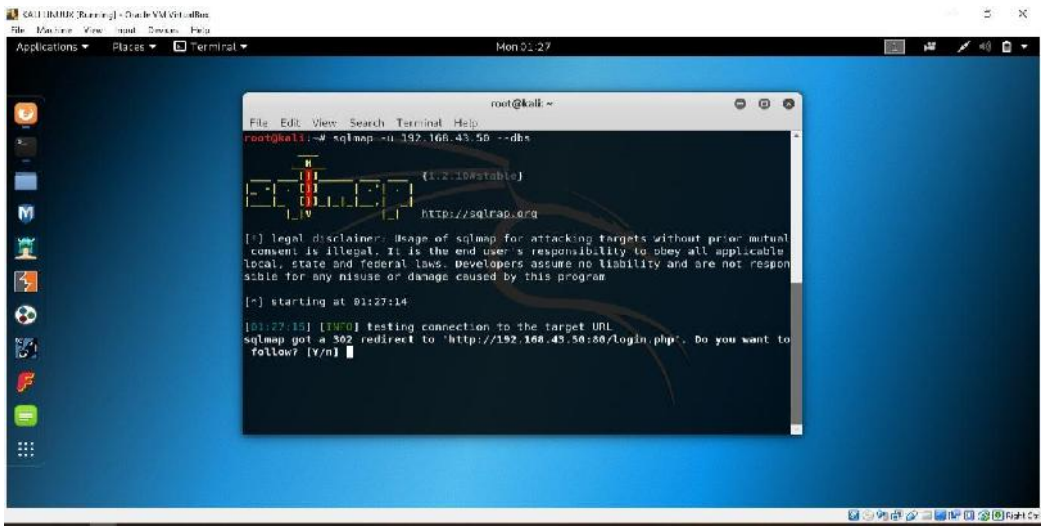


Gambar 4.23 Tampilan instalasi Kali linux yang telah berhasil

4.5 Pengujian Kali Linux

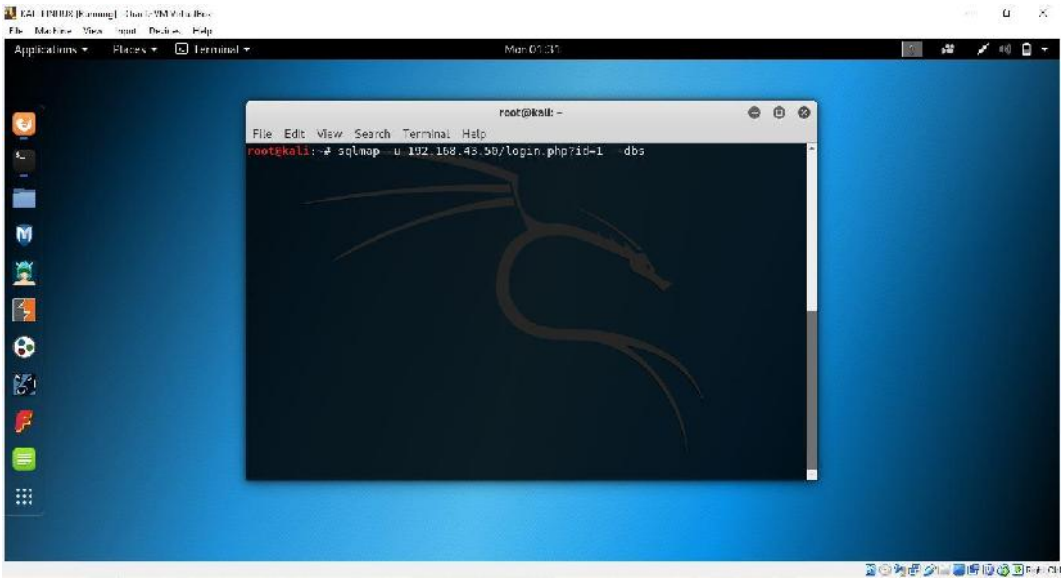
Gambar 4.24 Tampilan Pengujian kali linux

Pada Gambar 4.24 yaitu Setelah masuk pada Website LKP Multi Logika, Maka selanjutnya melakukan pengujian Kali Linux untuk melihat suatu pengamanan dimana sistem tersebut mempunyai celah masuk serangan atau tidak.



Gambar 4.25 Tampilan hasil Sqlmap

Pada Gambar 4.25 adalah Hasil dari pengujian Sqlmap –u 192.168.43.50 –dbs adalah Menandakan dimana website tersebut masih mempunyai keamanan dan tidak adanya celah serangan lain untuk masuk pada sistem tersebut.



Gambar 4.26 Tampilan Pengujian Kali linux

Pada Gambar 4.26 Dimana pengujian ini adalah untuk melihat dimana website tersebut mempunyai celah akan serangan atau tidak .

Gambar 4.27 Tampilan hasil pengujian serangan

Pada Gambar 4.27 adalah hasil pengujian pada website dengan sqlmap –u 192.168.43.50/login.php?id=1 –dbs , dimana website yang di tandai dengan php?id adalah website yang mempunyai celah untuk masuknya sebuah serangan ke dalam sebuah sistem tersebut.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil analisa keamanan *Server* pada pengujian keamanan dari serangan *SQL Injection* menggunakan system operasi kali linux di LKP Multi Logika Binjai, maka didapat beberapa kesimpulan seperti berikut:

a. Penerapan sistem keamanan *server* pada *login page web server* dengan enkripsi SHA 1 dari serangan *SQL Injection* pada LKP Multi Logika Binjai ini ditujukan untuk seluruh pegawai dan para peserta LKP agar dapat menggunakan internet secara aman tanpa adanya gangguan *attacker*.

b. Dalam melakukan pengujian kemanan *server* para pengguna di LKP Multi Logika Binjai dapat dengan mudah mengakses internet pada jaringan yang telah tersedia pada LKP untuk dipergunakan dalam proses belajar mengajar ataupun kepentingan lembaga lainnya.

5.2. Saran

Berikut adalah saran dari penulis agar analisa keamanan *server* pada pengujian keamanan dari serangan *SQL Injection* pada jaringan di LKP Multi Logika Binjai ini dapat bermanfaat dan dikembangkan menjadi lebih baik lagi :

a. Sistem yang telah dianalisa ini akan diimplementasikan pada LKP Multi Logika Binjai dalam hal pengujian keamanan *server* untuk

menghindari sejumlah serangan *attacker* seperti serangan *SQL Injection* yang dapat merusak sistem maupun melakukan pencurian data dari suatu sistem. Dalam menggunakan akses internet pada jaringan yang telah tersedia pada LKP. Untuk kedepannya, sistem yang dianalisa ini perlu diterapkan pada semua LKP yang ada agar para pegawai dan peserta les LKP dapat menggunakan internet dengan aman.

b. Dalam sistem yang dianalisa ini, penulis hanya menganalisa dengan menggunakan sistem operasi kali linux untuk melakukan pengujian keamanan *server* dari *attacker* dan untuk melakukan pengujian keamanan jaringan yang tersedia.

c. Untuk saat ini sistem yang dianalisa masih dalam pengujian dengan menggunakan *software VMware* untuk mensimulasikan sistem yang dianalisa oleh penulis.

DAFTAR PUSTAKA

Akbar, A. (2018). Pembangunan Model Electronic Government Pemerintahan Desa Menuju Smart Desa. Jurnal Teknik dan Informatika, 5(1), 1-5.

Ar, L. A., Prayudi, Y., & Yudha, F. (2018). *Portable Web Penetration Test Tool Memanfaatkan Single Board PC*. 42–48.

Batubara, S., Hariyanto, E., Wahyuni, S., Sulistianingsih, I., & Mayasari, N. (2019, August). Application of Mamdani and Sugeno Fuzzy Toward Ready-Mix Concrete Quality Control. In Journal of Physics: Conference Series (Vol. 1255, No. 1, p. 012061). IOP Publishing.

Dasar, A. K., & Jaringan, K. (2017). *Analisis Perbandingan Sistem Keamanan JaringanMenggunakan Snort dan Netfilter*. *5*(1).

Hardinata, R. S. (2019). Audit Tata Kelola Teknologi Informasi menggunakan Cobit 5 (Studi Kasus: Universitas Pembangunan Panca Budi Medan). Jurnal Teknik dan Informatika, 6(1), 42-45.

Harjono, E. B., Sarjana, P., & Ilmu, S. (2019). *Analisa Dan Implementasi Dalam Membangun Sistem Operasi Linux Menggunakan Metode LSF Dan REMASTER*. *1*, 30–35.

Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. TECHSI-Jurnal Teknik Informatika, 9(2), 103-122.

Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In Seminar Nasional Teknologi Informasi dan Multimedia (pp. 6-7).

Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. Journal of Saintech Transfer, 1(1), 27-32.

Herdianto, H. (2018). Perancangan Smart Home dengan Konsep Internet of Things (IoT) Berbasis Smartphone. Jurnal Ilmiah Core IT: Community Research Information Technology, 6(2).

Herdianto, H., & Anggraini, S. (2019, May). PERANCANGAN SISTEM PENDETEKSI UANG PALSU UNTUK TUNA NETRA MENGGUNAKAN ARDUINO UNO. In Seminar Nasional Teknik (SEMNASTEK) UISU (Vol. 2, No. 1, pp. 136-140).

Irawan, A. S., Pramukantoro, E. S., & Kusyanti, A. (2018). Pengembangan Intrusion Detection System Terhadap SQL Injection Menggunakan Metode Learning Vector Quantization. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *2*(6), 2295–2301.

Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. Jurnal Teknologi Informasi dan Pendidikan, 11(1), 1-6.

Kurnia, D. (2017). Analisis QoS Pada Pembagian Bandwidth Dengan Metode Layer 7 Protocol, PCQ, HTB Dan Hotspot Di SMK Swasta Al-Washliyah Pasar Senen. CESS (Journal of Computer Engineering, System and Science), 2(2), 102-111.

Kurniawan, F., Kusyanti, A., & Nurwarsito, H. (2017). Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *1*(9), 803–812. Retrieved from http://j-ptiik.ub.ac.id/index.php/j- ptiik/article/view/247

Lika, S., Dwi, R., Halim, P., & Verdian, I. (2018). *1 , 2 , 3*. *4*(2), 88–94.

Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). Int. J. Eng. Trends Technol, 38(7), 380-383.

Marlina, L., Putera, A., Siahaan, U., Kurniawan, H., & Sulistianingsih, I. (2017). Data Compression Using Elias Delta Code. Int. J. Recent Trends Eng. Res, 3(8), 210-217.

Muhammad Addy Rahmadani, Mochammad Fahru Rizal, T. G. (2017). Implementasi Hacking Wireless Dengan Kali Linux Menggunakan Kali Nethunter Wireless. *E-Proceeding of Applied Science : Vol.3, No.3 Desember 2017 | Page 1767 ISSN : 2442-5826*, *3*(3), 1767–1774.

Nasional, J., Informasi, S., Rahmatulloh, A., & Msn, F. (2017). *Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi*. *02*, 241–248.

Nurmalina, R. (2017). *Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut)*. *9*(1), 84–91.

Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. International Journal of Recent Trends in Engineering & Research, 3(8), 196-201.

Sandi, S., Putra, H., & Selatan, S. P. (2017). *Penanggulangan Serangan XSS , CSRF , SQL Injection Menggunakan Metode Blackbox Pada Marketplace IVENMU*. *4*(2), 289–300.

Simargolang, M. Y. (2017). *IMPLEMENTASI KRIPTOGRAFI RSA DENGAN PHP*. *1*, 1–10.

Soepomo, P. (2014). *ANALISIS DAN PERANCANGAN PROXY SERVER*. *2*, 1–9. Studi, P., Komputer, S., Teknologi, F., Universitas, I., Raya, S., Network, S., …

Pustaka, K. (2017). *PERANCANGAN DAN ANALISIS KEAMANAN JARINGAN NIRKABEL DARI SERANGAN DDOS (DISTRIBUTED DENIAL OF SERVICE) BERBASIS HONEYPOT*. *4*(2).

Sulistianingsih, I. (2019). Sistem Pendukung Keputusan Penentuan Menu Makanan Sehat untuk Pasien Rawat Inap. Jurnal Teknik dan Informatika, 6(1), 6-11.

Tasril, V., & Putri, R. E. (2019). Perancangan Media Pembelajaran Interaktif Biologi Materi Sistem Pencernaan Makanan Manusia Berbasis Macromedia Flash. Jurnal Ilmiah Core IT: Community Research Information Technology, 7(1).

Tedyyana. (2016). Membuat Web Server Menggunakan Dinamic Domain. *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, *7*, 1–10. Retrieved from https://ejurnal.unilak.ac.id/index.php/dz/article/view/178

Windu Gata, G. (2016). Pemodelan UML sistem informasi Monitoring Penjualan dan stok barang. *Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)*, *IV*(2), 107–116. https://doi.org/10.2135/cropsci1983.0011183X002300020002x

Yulianingsih, Y. (2017). Menangkal Serangan SQL Injection Dengan Parameterized Query. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *2*(1), 46–49. https://doi.org/10.26418/jp.v2i1.15507