**ANALISA PENERAPA N *DMARC* YANG DIINTEGRASIKAN DENGAN *ANTI SPAM* DAN *ANTI VIRUS* UNTUK PENGAMANAN *MAIL SERVER***

**SKRIPSI**



**Oleh:  
RUDI KURNIAWAN  
1710510157**

**ROGRAM STUDI ILMUKOMPUTER**

**FAKULTAS TEKNIK DAN KESEHATAN  
UNIVERSITAS BUMIGORA  
MATARAM  
2021**

# 

# **BAB I**

# **PENDAHULUAN**

## **Latar Belakang**

Perkembangan teknologi saat ini sudah begitu pesat sehingga teknologi dapat memudahkan pekerjaan manusia hampir di segala bidang, surat elektronik adalah salah satu dari kemajuan teknologi dalam bidang komunikasi sehingga fungsi dari surat dapat digantikan dengan adanya surat elektronik, efisiensi biaya dan waktu menjadi alasan yang membuat banyak orang beralih dari surat menuju surat elektronik.

Ditambahkan terkait pentingnya sebuah keamanan jaringan

Mengingat betapa pentingnya media komonikasi di zaman sekarang ini maka beberapa orang melakukan penelitian terutama di bidang keamanan jaringan. Naufal Hanif 2018, penerapan protokol DomainKeys Identified Mail dapat mencegah email spoofing dengan cara melakukan otentikasi menggunakan metode pencocokan private key dan public key (Asymmetric keys). Sedangkan penerapan protokol Sender Policy Framework dapat mencegah email spoofing dengan cara melakukan otorisasi menggunakan metode pencocokan alamat IP server pengirim. Hasil atau keluaran yang dicapai yaitu mail server dapat terhindar dari email spam, email spoofing, dan virus untuk memastikan keamanan dan kenyamanan pengguna email serta menghindari dampak kerugian yang dapat ditimbulkan oleh email spam, email spoofing, dan virus.. Andrian Maftuh Nadzifan, Farih Nazihullah 2018, pendeteksi spoofing pada email menggunakan penerapan DKIM, SPF dan DMARC yang pada penelitian di gunakan Sebuah metode untuk melakukan deteksi ¬diperlukan untuk melihat apakah sebuah email terindikasikan sebagai spoof atau tidak. Naufal Herdyputra Ardhi, 2020, Forensik email dengan metode Header Analysis dianggap efektif untuk melacak alamat IP pengirim email, namun hal ini tidak dapat melacak posisi pengirim email secara akurat. Mengintegrasikan email forensik klasik dengan data mining dari Twitter data stream telah terbukti efektif untuk mendapatkan informasi geografis dan memeperkecil luas dari seluas kota menjadi seluas lingkungan, yang sangat berharga bagi pihak berwajib dalam menghemat waktu dan juga usaha untuk mengadili pelaku tindak kejahatan cyber.

Dari kutipan di atas ada beberapa kekurangan seperti *DKIM* memiliki masalah yang tidak dapat menentukan apakah tanda tangan itu sah, Naoya Kitagawa, Toshiki Tanaka, Masami Fukuyama and Nariyoshi Yamai 2016. Pertimbangan ini lah yang membuat penulis untuk menerapkan *Protocol DMARC* yang berfungsi untuk mendeteksi email palsu dan memberi tahu pengguna tanpa *DKIM* tanda tangan dengan memanfaatkan *DMARC* dan menerapkan system itu mengirimkan hasil verifikasi *DMARC* ke penerima, *Baracuda Central* sebagai tools *anti spam* dan *spoofing* yang dapat melakukan otorisasi bukan hanya melalui alamat IP saja namu juga dapat melalui URL dan *antivirus ClamAV* untuk mengatasi virus yang sangat tidak diinginkan oleh pengguna maupun penyedia layanan email. Sistem pencegahan *email spam, spoofing*, dan *virus* diharapkan dapat mengurangi dampak kerugian yang diakibatkan oleh *email spam, spoofing,* dan *virus*.

*DMARC (Domain-based Message Authentication, Reporting and Conformance), dan BARACUDA CENTRAL dapat digunakan sebagai otentikasi dan otorisasi email sehingga email client akan terbebas dari tindakan spoofing. Penerapan Anti Spam dan Anti Virus ClamAV juga diperlukan agar email server terhindar dari email spam dan virus, metode yang diterapkan oleh Anti Spam dan Anti Virus ClamAV yaitu dengan melakukan pengecekan haeder, body, dan attachment email kemudian di sampaikan ke pengguna.*

Manfaat dari penerapan DMARC, BARACUDA CENTRAL, Anti Spam dan Anti Virus ClamAV adalah untuk mengoptimalkan system keamanan jaringan server mail, dengan cara memblokir surat elektronik yang dianggap sebagai spam atau virus, meningkatkan kualitas keamanan surat elektronik sehingga pengguna dapat terhindar dari aktifitas spoofing dan virus yang disisipkan melalui surat elektronik.

## **Perumusan Masalah**

Sesuai dari latar belakang yang telah dipaparkan di atas maka rumusan masalah yang akan dikaji adalah bagaimana menganalisa penerapan DMARC (Domain-based Message Authentication, Reporting and Conformance), BARACUDA CENTRAL Anti Spam, dan Anti Virus ClamAV pada mail server agar mail server dapat terhindar dari email spam, virus dan pengguna email dapat terhindar dari aktifitas spoofing.

* 1. **Batasan Masalah**

Batasan masalah yang digunakan dalam penyusunan skripsi ini untuk menjadikan pembahasan menjadi lebih terarah dan fokus adalah sebagai berikut:

1. Rancangan uji coba diimplementasikan menggunakan VPS yang disewa pada penyedia layanan VPS. Pada VPS akan dilakukan instalasi CentOS Web Panel, konfigurasi DNS server, konfigurasi Mail server, dan komputer client digunakan untuk mengakses Mail User Agent berbasis web (Zimbra).
2. Sistem operasi VPS yang digunakan adalah CentOS 7.3.1611.
3. Aplikasi yang digunakan untuk memudahkan instalasi dan konfigurasi server adalah CentOS Web Panel.
4. Aplikasi MTA yang digunakan adalah Postfix untuk mengirim email.
5. Aplikasi MDA yang digunakan adalah Dovecot untuk menerima email.
6. Aplikasi MUA yang digunakan adalah Roundcube sebagai aplikasi email di sisi pengguna.
7. Aplikasi DNS server yang digunakan adalah bind9 agar email server dapat diakses menggunakan nama domain
8. Aplikasi HTTP server yang digunakan adalah Apache agar Mail Transfer Agent berbasis web dapat diakses melalui browser.
9. Pengujian yang dilakukan dengan mengirim surat elektronik yang terindikasi sebagai spam, kemudian melakukan pengiriman email spoofing, dan email yang mengandung virus, serta mengecek header email sebelum dan setelah penerapan DMARC, BARACUDA CENTRAL, anti spam, dan anti virus.
10. Pengujian DMARC dan BARACUDA CENTRAL dilakukan dengan cara mengirim email spoofing menggunakan Emkei’s Fake Mailer kemudian email spoofing tersebut dikirim ke Gmail dan Yahoo! Mail.
11. Pengujian Anti Spam dan Anti Virus dilakukan dengan cara mengirim email spam dan email yang mengandung virus ke mail server.

## **Tujuan dan Manfaat Penulisan**

### **Tujuan**

Adapun tujuan dari penulisan skripsi ini adalah untuk menganalisa Pengujian Protocol DMARC, Anti Spam dan Anti Virus dilakukan dengan cara mengirim email spam dan email yang mengandung virus ke mail server.

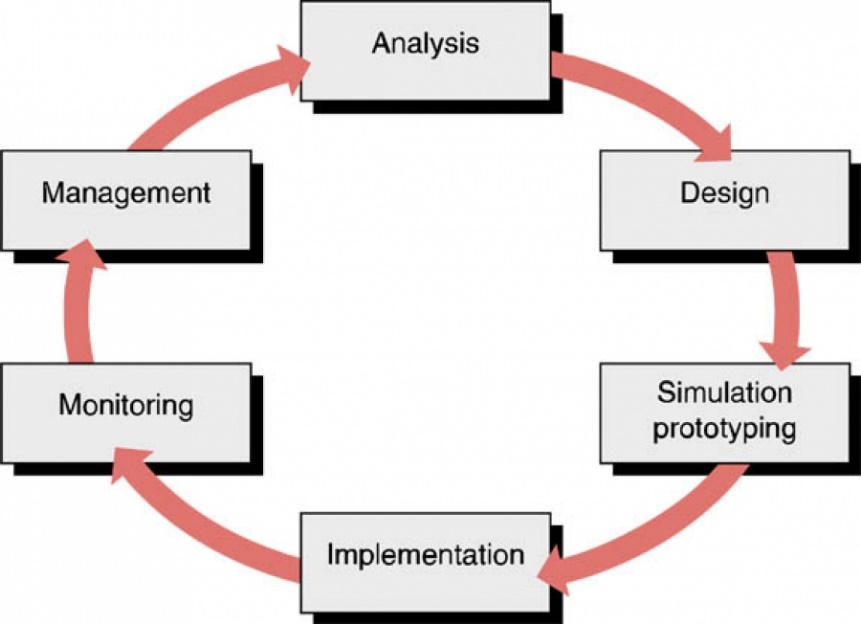
### **Manfaat**

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi Diri Sendiri
2. Dapat menambah ilmu pengetahuan pengetahuan baru yang dapat di terapkan di dunia kerja.
3. Dapat menjadi tempat untuk mengimplementasikan ilmu pengetahuan yang telah didapat selama berada dibangku perkuliahan.
4. Sebagai syarat untuk menyelesaikan jenjang pendidikan Strata 1 (S1) pada program studi Teknik Informatika di Universitas Bumigora Mataram.
5. Bagi Keilmuan
6. Dapat menjadi bahan rujukan untuk pengembangan penelitian berikutnya terutama dalam bidang yang sama.
7. Dapat menjadi sarana untuk melatih kemampuan dalam menulis karya ilmiah.
8. Bagi Masyarakat
9. Dapat memberikan pengetahuan terkait dengan analisa penanganan *email spam, virus* dan aktifitas *spoofing* menggunakan *Protocol DMARC*, *Baracuda Central*, *Anti Spam,* dan *Anti Virus*.
10. Dapat memberikan solusi cara penerapan *Protocol DMARC*, *Baracuda Central*, *Anti Spam,* dan *Anti Virus* pada *mail server*.

## **Metodologi penelitian**

Nurfajar, Kurniawan, dan Yunan (2015) menyebutkan bahwa *Network Development Life Cycle* adalah suatu metode yang digunakan dalam mengembangkan atau merancang jaringan infrastruktur yang memungkinkan terjadinya pemantauan jaringan untuk mengetahui statistik dan kinerja jaringan. *NDLC* mempunya enam fase, keenam fase tersebut dapat dilihat seperti pada gambar 1.1 berikut.



Gambar 1.1 Fase *NDLC*

Sumber: Nurfajar, Kurniawan, dan Yunan, 2015

Dari keenam fase yang terdapat pada *NDLC*, penulis hanya menggunakan lima fase antara lain sebagai berikut: (fase implementasi dan monitoring itu jika system kita diterapkan pada sebuah system real pada sebuah instansi) apakah system ini akan diterapkan ke suatu instansi real ?

1. *Analysis*

Pada fase ini penulis melakukan pengumpulan data dengan cara studi literatur, yaitu penulis membaca artikel ilmiah, buku, dan jurnal untuk mendapatkan informasi mengenai *DKIM*, *SPF*, *Anti Spam*, dan *Anti Virus*. Data-data yang telah terkumpul kemudian dianalisa.

1. *Design*

Pada fase ini penulis membuat rancangan yang meliputi rancangan jaringan uji coba, rancangan pengalamatan *IP*, rancangan sistem *filtering*, otentikasi, dan otorisasi *email* menggunakan *DMARC,*  *DKIM*, *SPF*, *Anti Spam*, dan *Anti Virus*, serta kebutuhan perangkat keras dan perangkat lunak.

1. *Simulation Prototyping*

Setelah melakukan analisa dan desain, tahap berikutnya adalah melakukan simulasi dan membuat *prototype* berdasarkan pada desain yang telah dirancang sebelumnya (Nurfajar, Kurniawan, dan Yunan, 2015). Pada fase ini dilakukan instalasi dan konfigurasi serta uji coba *DMARC,*  *DKIM*, *SPF* *Anti Spam*, dan *Anti Virus* menggunakan berbagai macam skenario.

1. *Implementation*

Pada fase ini penulis akan menerapkan semua yang telah direncanakan dan di desain pada tahapan sebelumnya. Pada fase ini penulis akan membangun sebuah *mail server* kemudian pada *mail server* tersebut akan diterapkan *DMARC* *DKIM*, *SPF*, *Anti Spam*, dan *Anti Virus* untuk mengotentikasi, mengotorisasi dan memvalidasi *email* dan penulis akan melakukan analisa pada *mail server* sebelum dan sesudah penerapan *DMARC,*  *DKIM*, *SPF*, *Anti Spam*, dan *Anti Virus*.

1. *Monitoring*

Setelah melakukan implementasi, tahapan *monitoring* adalah tahapan penting dalam merancang desain jaringan, tujuan dari tahapan *monitoring* adalah untuk memastikan jaringan komputer berjalan sesuai dengan tujuan pada tahap analisis (Nurfajar, Kurniawan, dan Yunan, 2015). Pada fase ini penulis akan melakukan *monitoring* terhadap aktifitas *spam*, dan *virus* pada *mail server* skripsian.online dengan menggunakan *maillog server* skripsian.online serta melakukan *monitoring* terhadap aktifitas *spoofing* pada penyedia layanan *email* yaitu *Yahoo! Mail* dan *Gmail*.

## **Sistematika Penulisan**

Adapun sistematika penulisan yang digunakan pada skripsi ini adalah sebagai berikut:

**BAB I Pendahuluan**

Bab ini berisi latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat penulisan, metodologi penelitian, dan sistematika penulisan.

**BAB II Landasan Teori**

Bab ini berisi tentang teori-teori yang melandasi penelitian ini, antara lain jaringan komputer, *OSI*, *TCP/IP*, keamanan jaringan komputer, tujuan keamanan jaringan komputer, *Spoofing*, *Phising*, *Server*, Linux, Linux *CentOS*, *CentOS Web Panel*, *Email*, *Mail Server*, *POP3*, *IMAP*, *SMTP*, *Postfix*, Dovecot, *Roundcube*, *DNS*, *DNS Server*, *BIND9*, *HTTP* dan *HTTPS*, *HTTP Server*, Apache *HTTP Server*, *Email Spam*, *Anti Spam*, *SpamAssassin*, *ClamAV*, *Amavisd-New*, *DKIM*, *OpenDKIM*, *SPF*, *Gmail*, *Emkei’s* *Mailer*, *Yahoo! Mail*.

**BAB III Metodologi Penelitian**

Bab ini berisi tentang metodologi penelitian yang digunakan dan fase-fase dari metodologi penelitian yang digunakan pada penelitian ini.

**BAB IV Hasil dan Pembahasan**

Bab ini berisi tentang pembahasan hasil konfigurasi, uji coba, dan analisa terhadap uji coba yang telah dilakukan.

**BAB V Penutup**

Bab ini berisi tentang kesimpulan dan saran untuk pengembangan skripsi ini selanjutnya.

# 

# **BAB II**

# **LANDASAN TEORI**

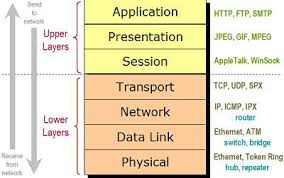
## **Jaringan Komputer**

Menurut Haryanto & Riadi (2014) sebuah jaringan komputer biasanya terdiri dari dua atau lebih komputer yang saling terhubung satu sama lain serta dapat saling berbagi sumber daya seperti *CDROM*, *printer*, pertukaran *file*, atau memungkinkan untuk saling berkomunikasi secara elektronik, komputer dapat terhubung melalui media transmisi seperti kabel, saluran telepon, gelombang radio, satelit atau infrared. Sedangkan menurut Masero, Triyono, dan Andayati (2013) jaringan komputer merupakan sekumpulan perangkat yang dapat digunakan untuk menyimpan dan memanipulasi data elektronis serta pesan-pesan, saling terkait sehingga dapat berbagi pakai berupa data, perangkat keras, dan perangkat lunak. Jaringan komputer adalah sekelompok komputer otonom yang saling berhubungan satu sama lain menggunakan protokol komunikasi sehingga dapat saling berbagi informasi, aplikasi, dan perangkat keras secara bersama-sama, tujuan membangun jaringan komputer adalah untuk membawa secara tepat tanpa adanya kesalahan dari sisi pengirim menuju ke sisi penerima melalui media komunikasi (Ardiantoro, Triyono, Fatkhiyah, 2016).

Berdasarkan dari ketiga pengertian tersebut dapat disimpulkan bahwa jaringan komputer adalah sebuah sistem yang menghubungkan *node-node* yang terdapat pada jaringan komputer dengan menggunakan media komunikasi tertentu sehingga *node-node* pada jaringan komputer dapat saling berbagi sumber daya, berkomunikasi, dan saling bertukar informasi untuk mencapai suatu tujuan yang sama.

## **Model Lapisan *OSI***

Model *Open System Interconnection* (*OSI*) diciptakan oleh *International Organitation for Standardization (ISO)* yang menyediakan kerangka logika terstruktur bagaimana proses komunikasi data berinteraksi melalui jaringan, standar ini dikembangkan untuk industri komputer agar komputer dapat berkomunikasi pada jaringan yang berbeda secara efisien (Kader, Najoan, dan Sinsuw, 2014).



Gambar 2.1 *Layer OSI*

Sumber: Sujana, 2014

Menurut Sujana (2014) terdapat tujuh *layer* pada model *OSI* dan setiap *layer* memiliki tanggung jawab khusus pada pada proses komunikasi data:

1. *Physical*

Pada *physical layer* tidak memiliki protokol yang spesifik, karena pada *physical layer* hanya mengirimkan *bit* data.

1. *Data Link*

Terdapat dua protokol pada *data link layer* yaitu:

* *PPP (Point to Point Protocol)*

Protokol yang digunakan untuk komunikasi *point to point* pada suatu jaringan.

* *SLIP (Serial Line Internet Protocol)*

Protokol yang digunakan untuk menghubungkan *serial*.

1. *Network*

Terdapat tiga protokol pada *network layer* yaitu:

* *IP (Internetworking Protocol)*

Mekanisme transmisi yang digunakan untuk mentransportasikan data dalam paket yang disebut *datagram*.

* *ARP (Address Resolution Protocol)*

Protokol yang digunakan untuk mengetahui alamat *IP* berdasarkan alamat fisik dari sebuah komputer.

* *RARP (Reverse Address Resolution Protocol)*

Protokol yang digunakan untuk mengetahui alamat fisik melalui alamat *IP* komputer.

* *ICMP (Internet Control Message Protocol)*

Mekanisme yang digunakan oleh sejumlah *host* untuk mengirim notifikasi datagram yang mengalamai masalah pada *host*nya.

* *IGMP (Internet Group Message Protocol)*

Protokol yang digunakan untuk memberi fasilitas pesan yang simultan kepada grup penerima.

1. *Transport*

Terdapat dua protokol pada *transport layer* yaitu:

* *TCP (Transmission Control Protocol)*

Protokol yang menyediakan layanan penuh pada lapisan *transport* untuk aplikasi.

* *UDP (User Datagram Protocol)*

Protokol *connectionless* dan *procces-to-procces* yang hanya menambahkan alamat *port*, *checksum error control* dan panjang informasi data pada *layer* diatasnya.

1. *Session*

Terdapat empat protokol pada *session layer* yaitu:

* *NETBIOS*

Berfungsi sebagai penyiaran pesan, maksudnya adalah memungkinkan *user* mengirim pesan tunggal secara serempak ke komputer lain yang terkoneksi. *NETBEUI (NETBIOS Extended User Interface)* berfungsi sama dengan *NETBIOS* hanya sedikit dikembangkan lagi dengan menambah fungsi yang memungkinkan bekerja dengan perangkat keras dan perangkat lunak.

* *ADSP (AppleTalk Data Stream Protocol)*

Fungsi dari protokol ini adalah untuk memantau aliran data dianatara dua komputer dan untuk memeriksa aliran data tersebut tidak terputus.

* *PAP (Printer Access Protocol)*

Berfungsi sebagai *printer postscript* untuk melakukan akses pada jaringan *Apple Talk* dan untuk mengendalikan bagaimana pola komunikasi antar *node*.

* *SPDU (Session Protocol Data Unit)*

Berfungsi sebagai penghubung antara dua *session service user*.

1. *Presentation*

Terdapat tiga protokol pada *layer presentation* yaitu:

* *TELNET*

Protokol yang digunakan untuk melakukan *remote access* ke suatu *host*.

* *SMTP (Simple Mail Transfer Protocol)*

Salah satu protokol yang digunakan dalam pengiriman *email* di internet atau untuk mengirim data dari komputer pengirim *email* ke *server email* penerima.

* *SNMP (Simple Network Management Protocol)*

Protokol yang digunakan dalam suatu manajemen jaringan.

1. *Application*

Terdapat sembilan protokol pada *layer application* yaitu:

* *HTTP (Hyper Text Transfer Protocol)*

Protokol yang digunakan untuk mentransfer dokumen dan web dalam sebuah *web browser* melalui *www*.

* *FTP (File Transfer Protocol)*

Protokol *internet* yang berjalan dalam lapisan aplikasi yang merupakan standar untuk mentransfer *file* komputer dalam sebuah jaringan *internet*.

* *NFS (Network File System)*

Jaringan komputer yang memungkinkan pengguna di klien komputer untuk mengakses *file* melalui jaringan dengan cara yang sama saat mengakses *file* pada sumber penyimpanan lokal.

* *DNS (Domain Name System)*

Protokol yang digunakan untuk memberikan suatu nama *domain* pada sebuah alamat *IP* agar lebih mudah diingat.

* *POP3 (Post Office Protocol)*

Protokol yang digunakan untuk mengambil *mail* dari suatu *mail transfer agent* yang akhirnya *mail* tersebut akan di*download* kedalam jaringan lokal.

* *MIME (*Multipurpose Internet Mail Extension*)*

Protokol yang digunakan untuk mengirim *file binary* dalam bentuk teks.

* *SMB (*Server Messange Block*)*

Protokol yang digunakan untuk mentransfer *server-server file* ke *DOS* dan Windows.

* *NNTP (Network News Transfer Protocol)*

Protokol yang digunakan untuk menerima dan mengirim *newsgroup*.

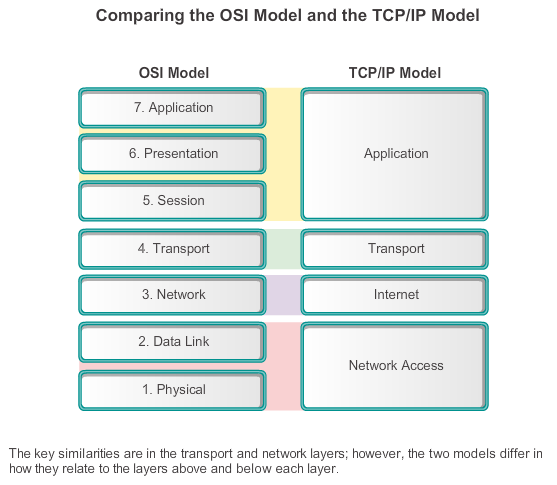
* *DHCP (Dynamic Host Configuration Protocol)*

Layanan yang memberikan alamat *IP* kepada komputer yang memintanya secara otomatis.

## ***Transmission Control Protocol / Internet Protocol (TCP/IP)***

*TCP/IP* didefinisikan sebagai protokol jaringan yang berperan dalam membangun *environment* jaringan global seperti internet. Protokol direferensikan pula sebagai suit protocol *DoD* (“*deeohdee*”) karena mereka pada dasarnya dikembangkan oleh komunitas riset *Advanced Research Projects Agency* (*ARPA*) dari *US Department of Defense* (*DoD*).

Nama *TCP*/*IP* diambil dari dua ‘Keluarga’ protokol fundamental, yaitu *TCP* dan *IP*. Meskipun demikian, suit masih memiliki protokol utama lainnya seperti *UDP* dan *ICMP*. Protokol bekerja sama dalam memberikan *framework networking* yang digunakan oleh banyak protokol aplikasi berbeda, di mana masing-masing digunakan untuk tujuan berbeda(Kader, Najoan, dan Sinsuw, 2014).



Gambar 2.2 Perbandingan *Layer TCP/IP* dan *Layer OSI*

Sumber: Wardoyo, Ryadi, dan Fahrizal, 2014

Berikut fungsi dari masing-masing *layer* pada protokol *TCP/IP* (Riadi, 2011):

1. *Network Access Layer*

*Layer network access* merupakan gabungan antara dua *layer* yaitu network interface layer dan physical layer, network interface layer berfungsi untuk mengirim data ke layer physical melalui device jaringan kemudian dilanjutkan oleh layer physical yang merupakan sistem kabel yang digunakan untukproses mengirim dan menerima data*.*

1. *Internet Layer*

Pada lapisan *internet* terjadi proses pengambilan paket dari lapisan transport dan menambahkan informasi alamat sebelum mengirimkannya ke lapisan network interface.

1. *Transport Layer*

Pada lapisan *transport* terdapat protokol seperti *TCP* dan *UDP* yang berfungsi menambahkan data *transport* ke paket dan melewatkannya ke lapisan Internet.

1. *Application Layer*

Pada lapisan *application* terdapat protokol seperti *FTP*, *Telnet*, *SMTP*, dan *NFS* dilaksanakan.

## **Keamanan Jaringan Komputer**

Menurut Fitriani (2014) Keamanan jaringan komputer merupakan suatu proses untuk mencegah dan mengidentifikasi penggunaan yang tidak sah dari jaringan komputer. Langkah-langkah pencegahan membantu menghentikan pengguna yang tidak sah yang disebut “penyusup” untuk mengakses setiap bagian dari sistem jaringan komputer. Tujuan keamanan jaringan komputer adalah untuk mengantisipasi resiko jaringan komputer berupa bentuk ancaman fisik maupun logik baik langsung ataupun tidak langsung mengganggu aktivitas yang sedang berlangsung dalam jaringan komputer.

## **Jenis-jenis Layanan Keamanan Jaringan**

Menurut Fitriani (2014) terdapat beberapa jenis layanan keamanan jaringan, diantaranya:

1. Otentikasi (*Authentication*)

Layanan Otentikasi ada 2 macam. Pertama disebut dengan Otentikasi Entitas (*Entity Authentication*) yaitu layanan kemanan jaringan yang memberikan kepastian terhadap identitas sebuah entitas yang terlibat dalam komunikasi data. Kedua adalah Otentikasi Keaslian Data (*Data Origin Authentication*) yaitu layanan yang memberikan kepastian terhadap sumber sebuah data.

1. Kendali Akses (*Access Control*)

Kendali Akses adalah layanan keamanan jaringan yang menghalangi penggunaan tidak terotorisasi terhadap sumber daya. Pada aplikasi jaringan umunya kebijakan kemampuan (baca, modifikasi, tulis dan eksekusi sebuah data/layanan sistem) ditentukan oleh jenis pengguna.

1. Kerahasiaan Data (*Data Confidentility*)

Kerahasiaan data adalah layanan keamanan jaringan yang memproteksi data tertranmisi terhadap pengungkapan oleh pihak yang tidak berwenang / berhak.

1. Keutuhan Data (*Data Integrity*)

Keutuhan data adalah layanan keamanan jaringan yang memastikan bahwa data yang diterima oleh penerima adalah benar-benar sama dengan data yang dikirim oleh pengirim.

1. *Non-Repudiation*

Layanan *non-repudiation* adalah layanan keaman jaringan yang menghindari penolakan atas penerima atau pengirim data yang telah dikirim.

1. Ketersediaan (*Availability*)

Layanan *Availability* adalah layanan sistem yang membuat sumber daya sistem tetap dapat diakses dan digunankan ketika ada permintaan dari pihak yang berwenang. Serangan seperti *Denial of Service* membuat sistem tidak dapat diakses oleh pihak yang berwenang.

## ***Email Spoofing* dan *Phising***

Menurut Suryana, Akbar, dan Widiyasono (2016) *Email Spoofing* adalah kegiatan melakukan manipulasi data pada *header email*. Serangan yang paling populer dari *email spoofing* adalah serangan *phising*. *Email spoofing* dianggap sebagai tindakan yang berbahaya, karena melakukan manipulasi data pada *header email* untuk menyamar sebagai orang atau organisasi yang berwenang, contohnya seperti melakukan pengiriman *email* dengan nama pengirim seolah-olah *email* tersebut dikirim oleh administrator suatu organisasi. Pengirim *email spoofing* menyerang dengan berbagai macam isi pesan untuk meyakinkan korbannya.

Menurut Suryana, Akbar, dan Widiyasono (2016) *Phising* adalah bentuk pencurian identitas secara *online* yang bertujuan untuk mencuri informasi sensitif seperti sandi dan informasi kartu kredit. Serangan *phising* menggunakan kombinasi teknik *social engineering* dan teknik *spoofing* untuk membujuk pengguna agar memberikan informasi sensitif yang dapat digunakan untuk memperoleh keuntungan pribadi, salah satu contohnya adalah keuntungan finansial. *Phiser* biasanya membajak sebuah halaman web dari bank, kemudian mengirim *email* kepada korbannya supaya korbannya mengunjungi situs berbahaya dengan tujuan untuk mengumpulkan informasi rekening bank dan nomor kartu milik korbannya.

## ***Server***

*Server* adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. Beberapa contoh layanan *server* adalah *DHCP Server*, *DNS Server*, *FTP Server*, *Web Server*, *Mail server*, *Database Server* dan lain-lain (Saputra & Syafrizal, 2012).

## **Linux**

Menurut Harjono (2016) 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 arsitektur Intel 80386. Banyak orang memiliki peran penting dalam mengembangkan dan memperluas Linux di berbagai belahan dunia. Peralatan sistem dan pustakanya umumnya berasal dari sistem operasi GNU yang diumumkan tahun 1983 oleh Richard Stallman. Kontribusi GNU merupakan dasar dari munculnya nama alternatif GNU/LINUX. Dia menggunakan alat proyek GNU dan dengan demikian sistem operasi dikembangkan melalui proyek GNU/LINUX.

## **Linux *CentOS***

*CentOS* merupakan singkatan dari Community ENTerprise Operating System yang merupakan sebuah distribusi Linux sebagai bentuk dari usaha untuk menyediakan *platform* komputasi berkelas *enterprise* yang memiliki kompatibilitas kode biner sepenuhnya dengan kode sumber yang menjadi induknya, *Red Hat Enterprise Linux (RHEL)*. *RHEL* merupakan distribusi Linux berbayar yang menyediakan akses update atas perangkat lunak dan beragam jenis dukungan teknis. Distribusi Linux ini sebenarnya merupakan gabungan dari sejumlah perangkat lunak yang didistribusikan di bawah lisensi perangkat lunak yang bebas dan kode sumber atas paket perangkat lunak ini dirilis ke publik oleh *Red Hat*. CentOS tersedia secara gratis, dukungan teknis utamanya disediakan terhadap para pengguna melalui mailing list, forum berbasis web, ataupun *chat*. Proyek CentOS tidak berafiliasi dengan *Red Hat*, sehingga proyek *CentOS* berjalan tanpa mendapatkan bantuan apapun dari *Red Hat*. Untuk penggalangan dana, *CentOS* berbasis donasi dari para pengguna serta sponsor dari perusahaan-perusahaan yang menggunakannya (Wicitra, Utomo, dan Wardana, 2014).

## ***Centos Web Panel***

*CentOS Web Panel* adalah panel kontrol untuk *web hosting* yang dapat digunakan secara gratis dan dirancang untuk memanajemen *VPS* maupun *Dedicated Server* dengan cepat dan mudah tanpa harus menggunakan aplikasi *SSH Client*, menawarkan sejumlah besar opsi dan fitur untuk manajemen *server* dalam paket panel kontrolnya (Control Web Panel, n.d.).

## **Surat Elektronik**

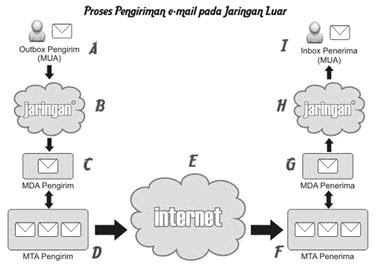
Surat elektronikadalah layanan yang diberikan oleh *internet* yang berkembang sejak tahun 1960, pada saat itu *internet* belum terbentuk, yang ada hanyalah kumpulan *mainframe* yang terbentuk sebagai jaringan. Mulai tahun 1980-an, surat elektroniksudah bisa dinikmati oleh khalayak umum. Surat elektronikadalah salah satu proses pengiriman surat melalui *internet* dengan menggunakan waktu yang sangat singkat.Surat elektronikmerupakan salah satu dari sekian banyak layananinternet yang ada saat ini selain *Netnews, Telnet, File Transfer Protokol (FTP) dan World Wide Web (www)* dan masih banyak layanan yang lainnya.Layanan internet adalah berbagai program atau fasilitas yang disediakan olehinternet, dari layanan internet tersebut yang paling banyak digunakan adalahlayanan surat elektronik. Penggunaan electronic mail (surat elektronik) sebagai media komunikasi yang ditunjang oleh banyaknya penyedia layanan di*internet* seperti Yahoo, Google, MSN, Wordpress, dan yang lainnya menunjukkan bahwa banyak orang melakukan komunikasi karena dengan komunikasi orang dapat beraktivitas dan meningkatkan kariernya (Mawarsih, 2014).

## ***Mail Server***

Menurut Desmira, Sumarto, dan Yuliani (2017) *Mail server* dikenal sebagai sebuah *mail transfer agent* atau *MTA, mail router* atau *mailer Internet* adalah sebuah aplikasi yang akan menerima *email* masuk dari pengguna lokal (orang-orangdalam satu domain) dan jarak jauh pengirim danmeneruskan *email* keluar untuk pengiriman. Sebuahkomputer yang didedikasikan untuk menjalankanaplikasi tersebut juga disebut sebagai *mail server. Mail Server* bisa diartikan sebagai induk atau rumahdari *email,* Setiap *email* yang dikirimkan dibuat untukmelewati sejumlah *server mail* sepanjang perjalanan kepenerima. Untuk *user* biasa, surat tersebut dikirimlangsung tetapi proses adalah sesuatu yangdimengerti. Tanpa rangkaian *Server Mail*, penggunahanya akan dapat mengirim email ke orang-orangyang memiliki alamat *email* dengan *domain* yang sama.

Menurut Muarif & Irwan (2017) *Mail Server* memiliki tiga komponen utama yangmembentuknya, yakni *Mail Transfer Agent (MTA),* Mail Delivery Agent *(MDA), dan* Mail User Agent *(MUA*):

1. Menurut Sadikin (2014) *Mail User Agent (MUA)* merupakan program yang digunakan oleh pemakai untuk membaca dan mengirim *email* pada komputer pribadinya. Contoh program atau perangkat lunak *Mail User Agent (MUA)* ini misalnya Microsoft Outlook, Microsoft Outlook Express, Lotus Notes, Pegasus Mail dan Thunderbird. *Mail User Agent (MUA)* mengambil *email* dari *email server* menggunakan protokol *Post Office Protocol (POP)* dan *Internet Message Access Protocol (IMAP)*.
2. *Mail Tranfer Agent (MTA) Mail Transfer Agent* merupakan salah satukomponen penting pada *server internet*. *Mail Transfer Agent* bertanggung jawab untuk mentransfer *email* dari *mail server* mengirimkan sampai ke *server* penerima *email*. Kebutuhan pengguna atas jenis *MTA* yang digunakan juga beragam. Berbagai kriteria biasadigunakan untuk pertimbangan. Tiap-tiap program *mail server* memiliki kelebihan dan kekurangantersendiri. Beberapa *MTA* memiliki fasilitas yangsangat hebat sehingga mampu digunakan untukmenangani email dalam jumlah ratusan bahkansampai ribuan perhari (Desmira, Sumarto, Yuliani, 2017).
3. Menurut Crocker (2009) *Mail delivery agent* atau *message delivery agent (MDA)* adalah komponen perangkat lunak komputer yang bertanggung jawab atas pengiriman pesan *email* ke kotak pesan penerima lokal.



**Gambar 2.3 Proses pengiriman *email***

Sumber: Pratama, 2008

Pada gambar 2.3 dapat dijelaskan proses pengiriman *email* dimulai dari proses A yaitu pengirim *email* mengirim *email* menggunakan *MUA*, kemudian *email* diteruskan pada *MDA* yang berfungsi untuk mengatur pengiriman *email* pada *mail server* lokal (proses C), jika *email* tersebut dikirim kepada penerima yang berada pada *mail server* yang berbeda maka *email* akan dikirim melalui *MTA* untuk diteruskan ke *mail server* penerima melalui jaringan *internet* (proses E) kemudian *email* tersebut diterima oleh *MTA* pada *mail server* penerima (proses F) dan dilanjutkan ke *MDA mail server* penerima (proses G) agar *email* dapat di unduh oleh penerima *email* melalui jaringan lokal (proses H dan I).

## ***Mail Protocol***

Menurut Desmira, Sumarto, dan Yuliani (2017) terdapat tiga *Mail Protocol*, yaitu:

1. *POP3 (Post Office Protocol version 3)*

*POP3* merupakan protokol yang digunakan untuk pengelolaan *email*. *POP3* memudahkan seseorang dalam mendapatkan *email* mereka dari sebuah *mail server* tanpa perlu koneksi yang lama dengan internet yang tentu saja memakan biaya.

1. *IMAP (Internet Message Access Protocol)*

*IMAP (Internet Message Access Protocol)* samahalnya dengan *POP3*, maka pesan *email* akansepenuhnya disimpan dalam *server email* danmenggunakan komputer lokal untuk mengirim danmengambilnya kapanpun di inginkan. Tergantung darikeinginan *user. IMAP* adalah *protocol* standar untuk mengakses ataumengambil *email* dari *server*.

1. *SMTP (Simple Mail Transfer Protocol)*

*SMTP* merupakan salah satu jenis *protocol* yangbekerja dalam hal pengiriman pesan-pesan berupa suratelektronik atau *email* pada sebuah jaringan *internet.*

## ***Postfix***

Menurut Hidayat (2010) “Postfix adalah *Mail Tranfer Agent* yang dapat diperoleh dengan gratis dan bersifat *open source*. *Postfix* merupakan *mail transfer agent default* untuk sejumlah sistem operasi yang bertipe unix. *Postfix* didistribusikan menggunakan lisensi umum *IBM* 1.0 yang merupakan lisensi perangkat lunak bebas tetapi tidak kompatibel dengan *GPL*”.

Menurut Kusmaya (2016) Postfix ditulis oleh Wietse Venema dan termasuk salah satu proyek freeware. Mulai digarap Wietse saat berkunjung ke *IBM* T. J. Watson *Research*. Wietse diberi kesempatan oleh IBM untuk menulis software ini. Original software tersebut diberi nama Vmailer, namun diganti menjadi *Postfix* atas saran *IBM*.

## **Dovecot**

Menurut Kusmaya (2016) “Dovecot adalah *open source server* *POP3* dan *IMAP* untuk Linux atau Unix. Program ini melengkapi *Postfix* dengan kinerja yang tinggi, kemudahan administrasi, dan keamanan yang solid. Dovecot merupakan sebuah aplikasi yang dijalankan untuk mengikuti protokol *IMAP* dan *POP3*.

## ***Roundcube***

*Roundcube* adalah solusi *webmail* gratis dan *open source* dengan antarmuka pengguna mirip *desktop* yang mudah dipasang atau dikonfigurasi dan berjalan pada *server LAMPP* standar. Tampilan menggunakan standar web terbaru untuk me*render* antar muka yang fungsional dan dapat disesuaikan. *Roundcube* menyertakan *library open-source* canggih lainnya seperti *PEAR*, *IMAP* yang berasal dari IlohaMail, pustaka Googiespell untuk pemeriksaan ejaan atau pembersih WasHTML oleh Frederic Motte (*Roundcube Open Source Webmail Software*, n.d.).

## ***Domain Name System (DNS)***

Menurut Saputra & Syafrizal (2012) *Domain Name System* adalah sebuah sistem yang menyimpan dan mengatur suatu informasi tentang penamaan *host* dari sebuah alamat *IP* menjadi sebuah karakter atau angka dalam sebuah jaringan internet yang di distribusikan pada *database*. *Domain name system* memiliki pengelolaan komponen inti yang terdiri dari *DNS resolver*, *Recursive DNS server* dan *Authoritative DNS server*. pada awal penggunaan *DNS* didalam jaringan komputer menggunakan HOSTS.TXT dari *SRI* (sekarang *SIR International*) yang berisi informasi dari nama komputer dan *IP address*.

## ***DNS Server***

Menurut Kusmaya (2016) *DNS server* adalah distribute database system yang digunakan untuk pencarian nama komputer di jaringan yang menggunakan *TCP/IP* (*Transmission Control Protocol/Internet Protocol*). *DNS server* biasa digunakan pada aplikasi yang terhubung ke *internet* seperti web browser atau email, dimana *DNS server* dapat membantu memetakan hostname sebuah komputer ke *IP* Address.

## ***Bind9***

*BIND9* adalah aplikasi *DNS server* yang paling umum digunakan di internet, khususnya di sistem unix, *bind9* merupakan standar *DNS server*. *BIND9* awalnya dibuat oleh empat orang mahasiswa dengan menggunakan CSRG di Universitas California, Berkeley dan pertama kali dirilis di dalam 4.3 BSD. Paul Vixie kemudian meneruskan pemrogramannya pada tahun 1988 saat bekerja di *DEC*. Saat ini, *Bind9* dikelola oleh Konsorsium sistem internet. *BIND9* awalnya di tulis pada awal 1980 dan didanai oleh *DARPA* (Defense AdvancedResearch Projects Agency). Pada pertengahan 1980-an, *DEC* (Digital EquipmentCorporation) mengambil alih pengembangan *BIND9*. Satu dari pekerja itu adalah Paul Vixie, yang terus mengerjakan *BIND9* sesudah meninggalkan *DEC* (Hidayat, 2010).

## ***HTTP***

Menurut Zabar dan Novianto (2015) *HTTP* adalah sebuah protokol yang bekerja dengan cara meminta atau menjawab antara *client* dan *server*. Sebuah *client HTTP* seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan *TCP/IP* ke *port* tertentu di tuan rumah yang jauh (biasanya *port* 80). Sebuah *server HTTP* yang mendengarkan di *port* tersebut menunggu *client* mengirim kode permintaan (*request*), seperti "GET / HTTP/1.1" (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan *MIME* yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut dan diikuti dengan badan dari data tertentu. Beberapa kepala (*header*) juga dapat ditulis atau tidak, sementara yang lainnya (seperti tuan rumah) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan bila ada), *server* mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta, atau sebuah pesan *error* atau pesan lainnya. Pengembangan *HTTP* dikoordinasi oleh Konsorsium *World Wide Web (W3C)* dan grup kerja *Internet Engineering Task Force (IETF)*, bekerja dalam publikasi satu seri *RFC*, yang paling terkenal *RFC 2616*, yang menjelaskan HTTP/1.1, versi *HTTP* yang umum digunakan sekarang.

## ***HTTP Server***

Menurut Syafrizal & Saputra (2012) *HTTP Server* adalah sebuah *software* yang melayani permintaan berupa *Hypertext Transfer Protocol (HTTP)* atau *Hypertext Transfer Protocol Secure (HTTPS)* dari komputer atau *client* yang terhubung dalam jaringan *internet* atau *intranet*.

## **Apache HTTP Server**

Menurut Syafrizal dan Saputra (2012) Apache HTTP Server adalah *web server* yang dapat dijalankan di banyak sistem operasi, seperti Unix, *BSD*, Linux, Microsoft Windows dan Novell Netware serta *platform* lainnya yang berguna untuk melayani dan memfungsikan situs web.

## ***Email Spam***

Menurut Chandra, Indrawan, dan Sukajaya (2016) *Spam email* dapat didefinisikan sebagai “*unsolicited bulk email*” yaitu *email* yang dikirimkan kepada ribuan penerima. *Spam email* biasanya dikirimkan oleh suatu perusahaan untuk mengiklankan produknya. Hal ini menyebabkan semakin padatnya antrian dari *mail server*. Banyak waktu yang terbuang untuk menghapus *email spam* dari kotak masuk, *spam* juga menyebabkan pemborosan biaya bagi pengguna yang menggunakan koneksi *dial-up*. Selain itu *spam* juga dapat membuang *bandwith* dan dapat menyebabkan penerima di bawah umur mengakses situs-situs yang memiliki konten negatif. Banyaknya *spam* menyebabkan kerugian dalam hal sumber daya dan memerlukan banyak waktu untuk menghapusnya.

## ***Spam Filter***

*Spam filter* merupakan *software anti spam*, *Software anti spam* bekerja dengan cara menganalisa *email* yang datang dan menggunakan sejumlah metode untuk menentukan apakah *email* yang diterima adalah *email spam* atau bukan. Keberhasilan *spam filter* dalam mencegah masuknya *email spam* tergantung dari *software anti spam* yang digunakan serta metode-metode yang diterapkan oleh *software anti spam* untuk mendeteksi dan mencegah *email spam* (Fachrurrazi, 2014).

## ***SpamAssassin, ClamAV, dan Amavisd-New***

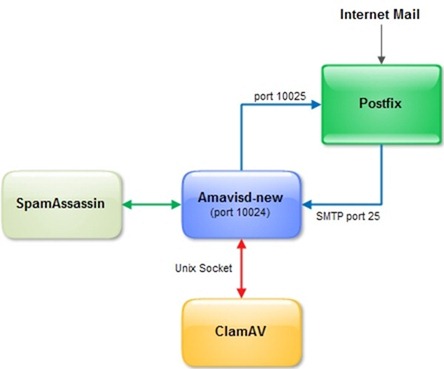
Menurut Irmayana & Nurlina (2014) *SpamAssassin* adalah aplikasi yang sudah teruji secara luas menggunakan proyek *open source* yang berfungsi sebagai *mail filter* untuk mendeteksi *spam*. *SpamAssassin* berjalan pada *server* dan sebagai *filter spam* sebelum sampai pada pada kotak masuk pengguna. *SpamAssassin* diintegrasikan dengan *mail server* agar secara otomatis menyaring semua *email spam* dan aturan penggunaan atau tes untuk menentukan *email spam* atau *ham*. *SpamAssassin* dapat memberikan tanda dengan mengubah *subject email* atau langsung menghapus *email spam* yang masuk.

*SpamAssassin* menggunakan berbagai mekanisme untuk menangani *email spam*, berikut mekanisme yang diterapkan SpamAssassin:

* + - 1. Pengecekan *header email*.
      2. Pengecekan isi *email*.
      3. Pengelompokan *email address* secara manual kedalam *whitelist* atau *blacklist*.
      4. *Bayesian filtering*.
      5. Penyaringan *database spam* kolaboratif (*DCC*, *Pyzor*, dan *Razor2*).
      6. Berbasis jaringan seperti *blacklist URL*, *blacklist DNS*, *checksum* berbasis *filter*, dan algoritma *Hash*.

Menurut Kusmaya (2016) *ClamAV* adalah *anti virus* open source *(GPL)* yang dirancang untuk mendeteksi trojan, *virus*, malware, dan ancaman berbahaya lainnya. Secara de facto *ClamAV* adalah standar untuk pemindaian mail gateway.

*Amavisd-new* adalah antarmuka yang memiliki kinerja yang tinggi dan dapat diandalkan. *Amavisd-new* memiliki beberapa fitur seperti pemindai *virus* dan modul *SpamAssassin*. *Amavisd-new* berkomunikasi ke *MTA* melalui protokol *SMTP* atau *LMTP*, atau dengan menggunakan program pembantu (Martinec, 2016).



Gambar 2.4 Cara Kerja *SpamAssassin*, *ClamAV*, dan *Amavisd-New*

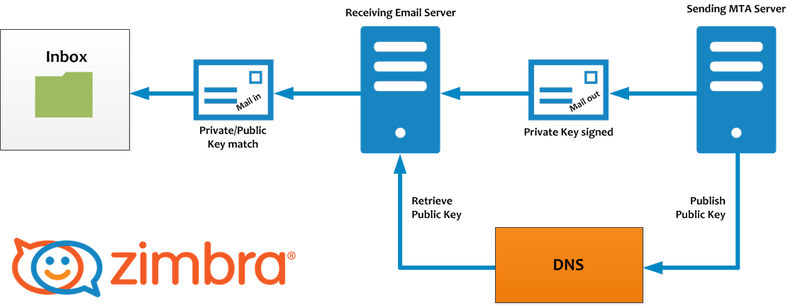
Sumber: [Valsecchi](https://nolabnoparty.com/en/author/paolo-valsecchi/), 2013

Cara kerja *SpamAssassin*, *ClamAV*, dan *Amavisd-New* dapat dilihat seperti gambar 2.4 yaitu *Amavisd-New* menerima *email* dari *Postfix* (*MTA*), kemudian menyebarkannya ke *ClamAV* dan *SpamAssassin* untuk memeriksa *spam* dan *virus* lalu mengembalikan *email* ke *Postfix* (*MTA*) untuk diteruskan ke penerima *email*.

## ***DomainKeys Identified Mail (DKIM)* dan *OpenDKIM***

*Domain Keys Identified Mail (DKIM)* adalah metode otentikasi *email* yang dirancang untuk mendeteksi *spoofing email*. Ini memungkinkan penerima untuk memeriksa bahwa *email* yang diklaim berasal dari *domain* tertentu memang diotorisasi oleh pemilik domain tersebut. Hal ini dimaksudkan untuk mencegah alamat pengirim palsu dalam *email* yang sering digunakan untuk melakukan *phishing* dan *spam email*. (Hansen, Crocker, Baker, 2009).

Menurut Barovih (2011) *OpenDKIM* adalah pengiriman *email* yang menggunakan mekanisme otentikasi *framework* menggunakan kunci publik yang dimasukan ke dalam *DNS* maupun *email*.



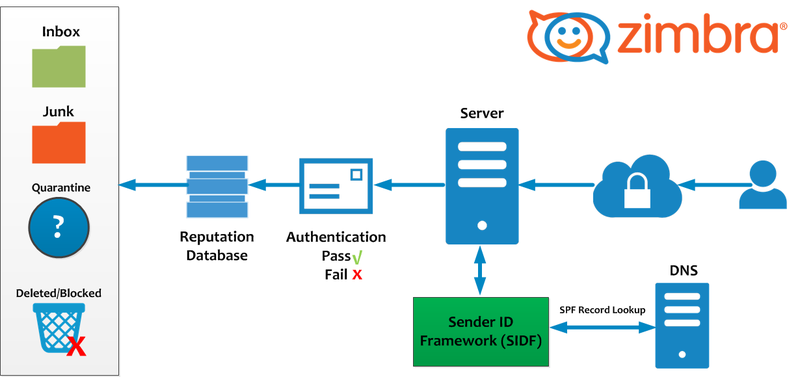
Gambar 2.5 Cara Kerja *DKIM*

Sumber: *Zimbra Incorporation*, 2005

Cara kerja *DKIM* dapat dilihat seperti pada gambar 2.5 yaitu *mail server* pengirim mem*publish public key* pada pada *DNS server* pengirim, setiap *email* yang dikirim melalui *mail server* pengirim akan diberikan *private key*, setelah *email* sampai pada *mail server* penerima maka *mail server* penerima akan mencocokan *private key* yang terdapat pada *email* dengan *public key* yang terdapat pada *DNS server* penerima, jika *public* dan *private key* cocok maka *email* tersebut dapat dipastikan berasal dari pengirim yang asli, namun jika *public* dan *private key* tidak cocok maka *email* tersebut dapat dipastikan sebagai *email spoofing*.

## ***Sender Policy Framework (SPF)***

*Sender Policy Framework (SPF)* adalah sistem validasi *email*, yang dirancang untuk mencegah *email* yang tidak diinginkan menggunakan sistem *spoofing*. Untuk memeriksa masalah keamanan umum ini, *SPF* akan memverifikasi *IP* sumber *email* dan membandingkannya dengan data TXT *DNS* dengan konten *SPF* (*Zimbra Incorporation, Best Practices on Email Protection: SPF, DKIM and DMARC*, 2005).



Gambar 2.6 Cara Kerja *SPF*

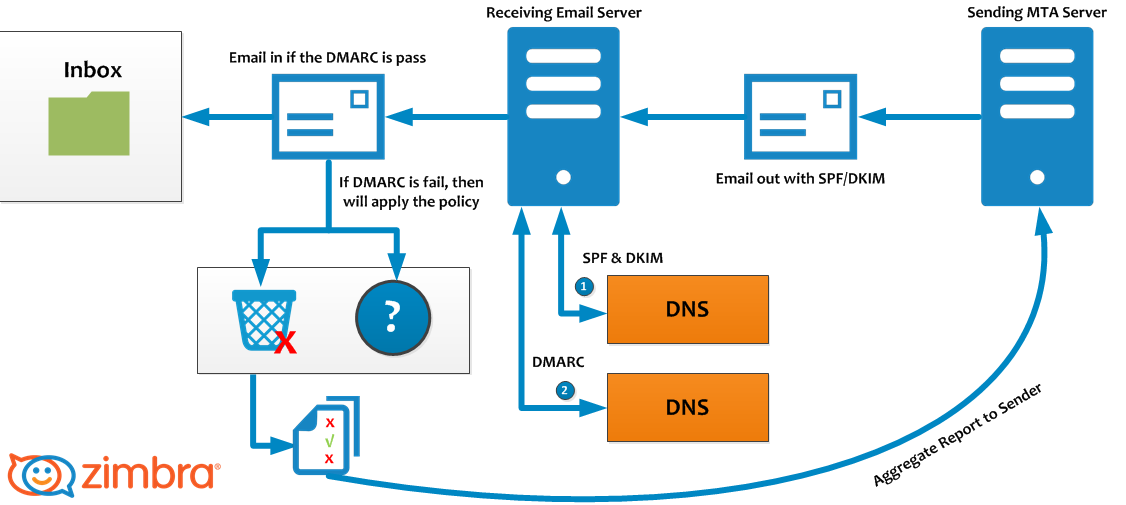
Sumber: *Zimbra Incorporation*, 2005

Cara kerja *SPF* dapat dilihat seperti pada gambar 2.6 yaitu *email* yang dikirim oleh pengirim akan diteruskan pada *mail server* penerima, selanjutnya *mail server* penerima akan mengecek *Sender ID Framework* yang berada pada *DNS* server pengirim, jika alamat *IP* *server* pengirim *email* sesuai dengan alamat *IP* yang telah diotorisasi oleh *SPF record* pada *DNS server* pengirim *email* maka *email* tersebut akan diberi nilai *PASS*, namun jika alamat *IP server* pengirim *email* tidak sesuai dengan alamat *IP* yang telah diotorisasi oleh *SPF record* pada *DNS server* pengirim *email* maka *email* tersebut akan diberi nilai *FAIL* atau *SOFTFAIL* dan selanjutnya *database* reputasi akan memberi nilai pada email tersebut berdasarkan pada laporan *SPF* masing-masing *email* untuk dijadikan pertimbangan tindakan apa yang akan dilakukan pada *email* tersebut.

## ***Domain-Based Message Authentication, Reporting & Conformance (DMARC)***

Domain-based Message Authentication, Reporting, and Conformance (DMARC) adalah metode autentikasi email standar. DMARC membantu administrator organisasi mencegah peretas dan penyerang lain melakukan spoofing terhadap organisasi dan domain. Spoofing adalah jenis serangan yang memalsukan alamat Dari dalam pesan email. Pesan palsu seolah tampak berasal dari organisasi atau domain yang ditiru identitasnya.

DMARC juga memungkinkan Anda meminta laporan dari server email yang menerima pesan dari organisasi atau domain Anda. Laporan ini berisi informasi untuk membantu Anda mengidentifikasi kemungkinan masalah autentikasi dan aktivitas berbahaya untuk pesan yang dikirim dari domain Anda.



Gambar 2.7 Cara Kerja *DMARC*

Sumber: *Zimbra Incorporation*, 2005

DMARC melalukan pengecekan melalui mekanisme yang sama dengan SPF Record dan DKIM Record. Saat pengguna mengirim email (melalui server indoglobal.com), maka mail server akan membubuhkan tandatangan yang dapat dicek oleh mail server penerima untuk memastikan keabsahan email tersebut.

Mail server penerima akan melakukan pengecekan melalui dua cara:

* 1. Dengan mekanisme SPF, yaitu dengan cara melihat IP address mail server yang mengirimkan email. Jika IP address tersebut tertera pada record SPF, maka email tersebut dianggap sah.
  2. Dengan mekanisme DKIM, yaitu dengan cara melihat tanda tangan yang ada di email. Jika tanda tangan di email sesuai dengan yang dipublikasikan pada record DKIM, maka email dianggal sah.

Pengecekan DMARC dianggap sah jika paling tidak SALAH SATU dari kondisi di atas terpenuhi.

## ***Gmail***

*Gmail* adalah layanan *email* yang intuitif dan efisien. *Gmail* menyediakan penyimpanan sebesar 15 GB, dengan lebih sedikit *spam*, dan dapat diakses melalui perangkat seluler (Gmail .n.d).

## ***Emkei’s* *Mailer***

*Emkei’s* *Mailer* adalah *Mailer* palsu *online* gratis dengan berbagai fitur seperti lampiran, enkripsi, *Editor HTML*, dan pengaturan lanjutan. (Emkei’s Mailer, 2009).

*Emkei’s* *Mailer* dapat digunakan untuk mengirim *email spoofing* dengan memalsukan alamat *email* pengirim pesan. *Emkei’s* *Mailer* dapat diakses menggunakan *browser* dengan alamat *domain* www.emkei.cz. *Emkei’s* *Mailer* dapat diakses secara gratis sehingga memberikan kemudahan dalam mengirim *email spoofing*.

## ***Yahoo! Mail***

*Yahoo! Mail* merupakan sebuah penyedia surat elektronik (*webmail*) dari [Yahoo!](https://id.wikipedia.org/wiki/Yahoo!). *Yahoo! Mail* merupakan penyedia surat elektronik terbesar di [*internet*](https://id.wikipedia.org/wiki/Internet) dengan jutaan pengguna. Saingan utama *Yahoo! Mail* ialah [*Windows Live Hotmail*](https://id.wikipedia.org/wiki/Windows_Live_Hotmail), [*Gmail*](https://id.wikipedia.org/wiki/Gmail) dan [*AOL Mail*](https://id.wikipedia.org/wiki/AOL_Mail) (Arrington, 2006).

# 

# **BAB III**

# **METODOLOGI DAN PERANCANGAN**

Metode penelitian yang digunakan adalah *Network Development Life Cycle (NDLC).* Dari enam tahapan yang ada pada *NDLC*, penulis hanya menggunakan 5 tahapan yaitu *Analysis, Design, Simulation Prototyping,* *Implementation,* dan *Monitoring.*

## **Tahap Analisa (*Analysis*)**

Pada fase ini penulis melakukan pengumpulan data dengan cara studi literatur, yaitu penulis membaca artikel ilmiah, buku, dan jurnal untuk mendapatkan informasi mengenai *email spam*, *email spoofing*, dan *virus*. Data-data yang telah terkumpul kemudian dianalisa. Tahap ini terdiri dari dua bagian yaitu pengumpulan data dan analisa data.

### **Pengumpulan Data**

Pada tahap pengumpulan data, penulis menggunakan metode studi literatur yaitu dengan mempelajari beberapa jurnal ilmiah yang membahas tentang *email spam, email* *spoofing*, dan *virus*, selain itu penulis juga menggunakan *e-book* yang membahas tentang *email spam, virus,* dan *email* *spoofing*. Setelah membaca beberapa jurnal ilmiah diperoleh informasi tentang beberapa jurnal ilmiah yang berkaitan dengan *email spam, virus,* dan *email* *spoofing* seperti terlihat pada tabel 3.1 berikut.

Tabel 3.1 Jurnal Ilmiah Tentang *Email Spam, Spoofing,* dan *Virus*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Penulis** | **Tahun** | **Judul** | **Pembahasan** |
| 1 | Andri Lesmana Suryana, R. Reza El Akbar, dan Nur Widiyasono | 2016 | Investigasi *Email Spoofing* dengan Metode *Digital Forensics Research Workshop (DFRWS)* | Mengidentifikasi *email* *spoofing* menggunakan metode *DFRWS* |
| 2 | Naufal hanif.S.Kom | 2018 | Analisa Penerapan DomainkeysIdentifiedMail(Dkim), SenderPolicyFramework(Spf), AntiSpam, Dan AntiVirus Pada Mail Server | menganalisa penerapan *DKIM*, *SPF*, *Anti Spam* dan *Anti Virus* pada *mail server* agar *mail server* terhindar dari *email spam*, *virus* dan aktifitas *spoofing* |
| 3 | Andrian Maftuh Nadzifan, Farih Nazihullah | 2018 | Aplikasi Untuk Deteksi Adanya Spoof Pada *Email* | Meneruskan penelitian sebelumnya *Forensic* Analysis of E-mail Address Spoofing demgan algoritma deteksi. |
| 4 | Abidarin Rosidi, Heri Sismoro, Emha Taufiq Luthfi,  Hanif Al Fatta, Hastari Utama | 2016 | Data Manajemen Dan Teknologi Informasi | mendeteksi adanya emailspoofing, maka perlu adanya investigasi forensik email terhadap emailspoofing. |
| 5 | Daniel Adi Putra Sitorus, Harun Mukhtar, Yulia Fatma | 2020 | Analisa Dan Implementasi Security Mail Server | Analisis dan implementasi serangan email spam pada mail server zimbra. |

### **Analisa Data**

Berdasarkan hasil dari pengumpulan data maka dapat diperoleh hasil analisa sebagai berikut:

1. Jurnal ilmiah pertama membahas tentang investigasi *email spoofing* menggunakan metode *DFRWS* yaitu dengan melakukan pengecekan *header email* secara manual.
2. Jurnal ilmiah kedua pembahasan tentang penerapan protokol *DomainKeys Identified Mail* dapat mencegah *email spoofing* dengan cara melakukan otentikasi menggunakan metode pencocokan *private key* dan *public key* (*Asymmetric keys)*. Sedangkan penerapan protokol *Sender Policy Framework* dapat mencegah *email spoofing* dengan cara melakukan otorisasi menggunakan metode pencocokan alamat *IP server* pengirim. Sebaliknya penerapan *SpamAssassin*, *ClamAV*, dan *Amavisd-New* dapat mencegah masuknya *email spam* dan *virus* dengan cara melakukan pengecekan *header*, *body*, dan *attachment email*.
3. Jurnal ilmiah ketiga membahas tentang Meneruskan penelitian sebelumnya *Forensic* Analysis of E-mail Address Spoofing demgan algoritma deteksi.
4. Jurnal ilmiah keempat membahas tentang mendeteksi adanya email spoofing, maka perlu adanya investigasi forensik email terhadap email spoofing.
5. Jurnal ilmiah kelima membahas tentang Analisis dan implementasi serangan email spam pada mail server zimbra.
6. Penanganan *email spoofing* belum menerapkan metode otentikasi dan otorisasi untuk menambah informasi pada *email header*.
7. Belum terdapat uji coba *ClamAV* sebagai *anti virus* pada *mail server*.

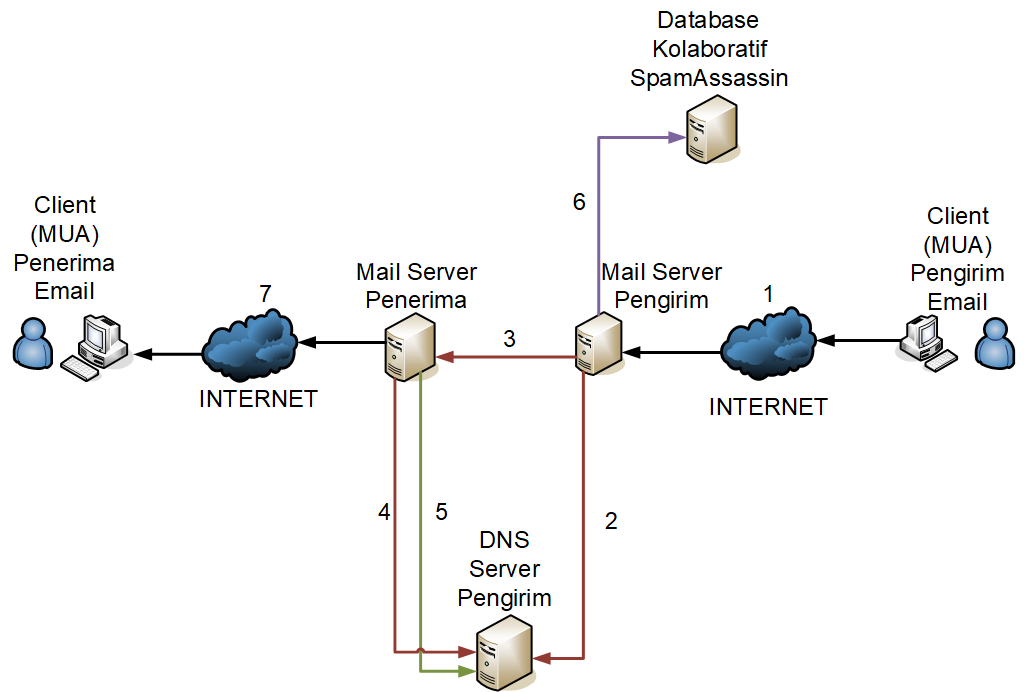
Dari hasil analisa tersebut maka mendorong penulis untuk melakukan penelitian tentang Analisa penerapan *Domainkeys identified mail (DKIM), sender policy framework (SPF) Anti Spam*, dan *Anti Virus* pada *Mail Server.*

## **Tahap Desain *(Design)***

Tahap ini terdiri dari 4 (empat) bagian yaitu rancangan sistem *filtering email spam, virus,* dan *spoofing*, rancangan jaringan ujicoba, rancangan pengalamatan *IP*, rancangan akun *email*, serta kebutuhan perangkat keras dan perangkat lunak.

### **Rancangan Sistem *Filtering Email Spam, Virus* dan *Spoofing***

Rancangan sistem *filtering email spam, virus* dan *spoofing* yang digunakan seperti terlihat pada gambar 3.1 berikut.



Gambar 3.1 Rancangan Sistem *Filtering Email Spam,* *Spoofing,* dan *Virus*

Berdasarkan gambar 3.1 tersebut maka rancangan sistem *filtering email spam*, *virus* dan *email* *spoofing* dapat dijelaskan sebagai berikut.

* + - * 1. Langkah 1 *user* mengirim *email* dengan menggunakan *Mail User Agent* berbasis web (*Roundcube*), *user* mengakses *Roundcube* menggunakan *browser.*
        2. Langkah 2 *Mail server* pengirim meneruskan *email* ke *mail server* penerima dengan menambahkan *private key* pada *header email*.
        3. Langkah 3 *Mail server* pengirim mem*publish public key* pada *DNS server*nya.
  1. Langkah 4 *Mail server* penerima mengambil *public key* yang ada pada *DNS server* pengirim *email* untuk dicocokan dengan *private key* yang ada pada *header email*, jika *private key* tidak cocok dengan *public key* maka *email* akan dianggap sebagai *email spam*, jika *private key* cocok dengan *public key* maka proses akan berlanjut pada langkah ke 5.
  2. Langkah 5 *Mail server* penerima mencocokan alamat *IP mail server* pengirim dengan *sender ID framework* pada *SPF record* yang berada pada *DNS server* pengirim, jika pada *SPF record* yang berada pada *DNS server* pengirim tidak mengotorisasi alamat *IP email server* pengirim *email* tersebut maka *email* tersebut akan diblok atau ditandai sebagai spam, jika alamat *email* pengirim telah diotorisasi oleh *administrator email server* maka proses akan berlanjut pada proses ke 6.
  3. Langkah 6 *Mail server* penerima melakukan pengecekan pada *database* kolaboratif *SpamAssassin*.
  4. Proses pem*filter*an *email* *spam* selanjutnya adalah menggunakan *SpamAssassin* dan *ClamAV* sebagai *anti spam* dan *anti virus* *email* dengan *Amavisd-New* sebagai penghubung antara *SMTP server* dengan *SpamAssassin* dan *ClamAV*.

Keterangan: garis merah mewakili proses *DKIM* (nomor 2, 3, dan 4), garis hijau mewakili proses *SPF* (nomor 5), dan garis ungu mewakili proses *SpamAssassin* (nomor 6).

### **Rancangan Jaringan Uji Coba**