BAB II

LANDASAN PEMIKIRAN

2.1. World Wide Web

World wide web (www) merupakan sekumpulan *web server* dari berbagai penjuru yang berisi data dan informasi untuk di sebarluaskan ke se seluruh dunia dengan tujuan bersama. Dalam sejarahnya www adalah suatu program yang ditemukan oleh Tim Barnerrs-Lee pada tahun 1991 [WIK10]. Awalnya Berners-Lee hanya ingin menemukan cara untuk menyusun arsip-arsip risetnya.Untuk itu, dia mengembangkan suatu sistem untuk keperluan pribadi. Sistem itu adalah program peranti lunak yang diberi nama *Equire*. Dengan program itu, Berners-Lee berhasil menciptakan jaringan terkait antara berbagai arsip sehingga memudahkan informasi yang dibutuhkan.Inilah yang kemudian menjadi dasar dari sebuah revolusi yang dikenal sebagai web.

WWW dikembangkan pertama kali di Pusat Penelitian Fisika Partikel Eropa (CERN), Jenewa, Swiss.Pada tahun 1989 Berners-lee membuat proposal untuk proyek pembuatan *hypertext* secara global, kemudian pada bulan Oktober 1990, 'World Wide Web' sudah bisa dijalankan dalam lingkungan CERN. Pada musim panas tahun 1991, WWW resmi digunakan secara luas pada jaringan Internet [WP10].

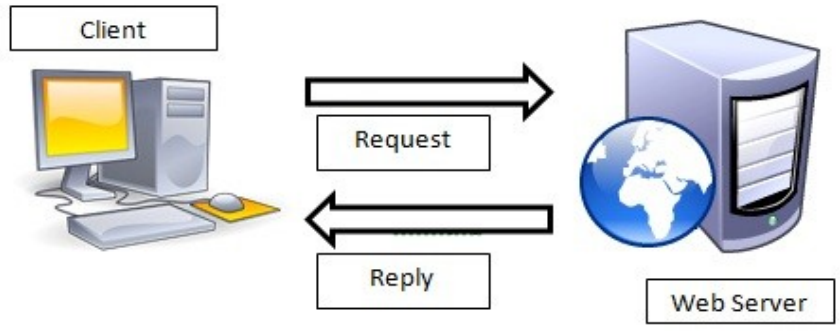
2.1.1. Web Portal

Web portal merupakan seluruh kumpulan informasi, dokumentasi yang tergabung dalam sebuah alamat web yang mempunyai nama domain tertentu. Dalam website portal tersebut kita bisa melihat informasi terkait secara lengkap dan tergantung kebutuhan informasi yang kita perlukan. Sepertihalnya kita sering membaca berita yang selalu diupdate setiap detiknya kita bisa mengunjungi web site portal berita yang beralamat di http://www.detik.com. Jika web portal tersebut

berisi tentang informasi kebijakan suatu pemerintah, kemajuan dan visi dan misi pemerintah itu merupakan web site portal milik pemerintah, jadi terlihat perbedaannya dari isi konten web tersebut. Seperti yang kita ketahui sebuah web portal instansi pemerintah bidang pertanian yang memiliki domain http://www.deptan.go.id

2.1.2. Web Server

Web server merupakan perangkat lunak yang menangani permintaan dari *client* berupa halaman web yang diterjemahkan oleh perangkat lunak tersebut secara *realtime* melalui perantara *web browser* seperti : *mozilla firefox, opera mini, netscape, google chrome* maupun *internet explorer*. Kita mengenal beberapa *web server* versi *open source* seperti : *Apache , Nginx, Tomcat*. Adapun *web server* yang sifatnya *propietary* yang kita kenal saat ini adalah *Microsoft IIS*. Semua *web server* itu pada prinsipnya sama sebagai layanan komunikasi antara web aplikasi dan *database* aplikasi yang tersimpan dalam *web server* tersebut. Berikut ilustrasi cara kerja sebuah *web server* :



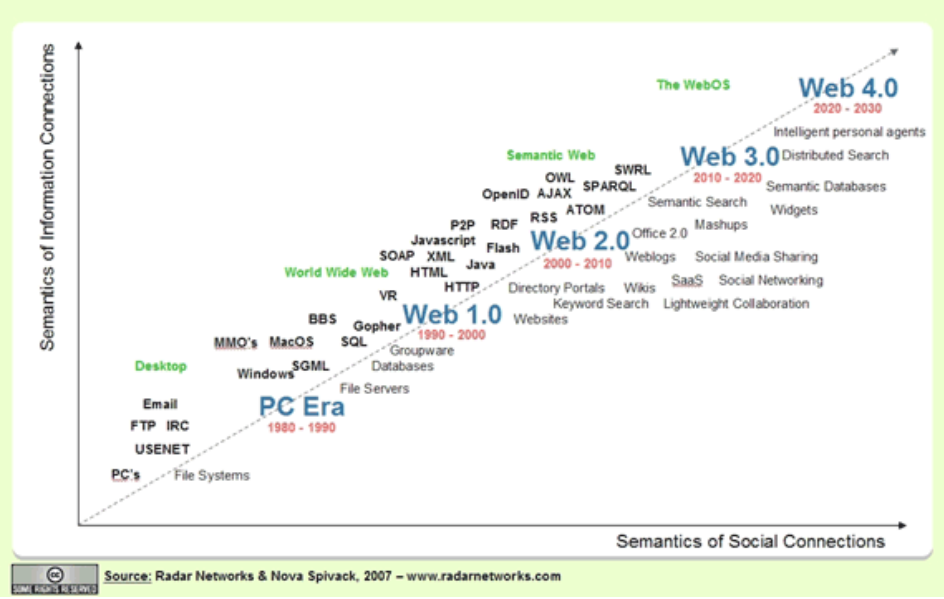
Gambar II.1: Cara Kerja Web Server [BUG11]

Sebuah *client* melalui *web browser* meminta halaman web kepada alamat website tersebut lalu di kirim permintaan tersebut ke *web server* dan di kembalikan permohonan halaman web client tersebut sesuai pesanan. Teknologi web saat ini sudah memasuki era teknologi web 3.0 dimana aplikasi web tersebut

tidak hanya lagi dinamis dalam implementasinuya, tetapi melainkan semantik *database*, yaitu Web semantik merujuk kepada kemampuan aplikasi komputer untuk lebih memahami bahasa manusia, bukan hanya bahasa yang baku dari para penggunanya tetapi juga bahasa yang lebih kompleks, seperti dalam bahasa percakapan sehingga memudahkan penggunanya untuk berkomunikasi dengan mesin. Web semantik dapat mengolah bahasa dan mengenali homonim, sinonim, atau atribut yang berbeda pada suatu *database*.

2.1.3. Teknologi Perkembangan Web

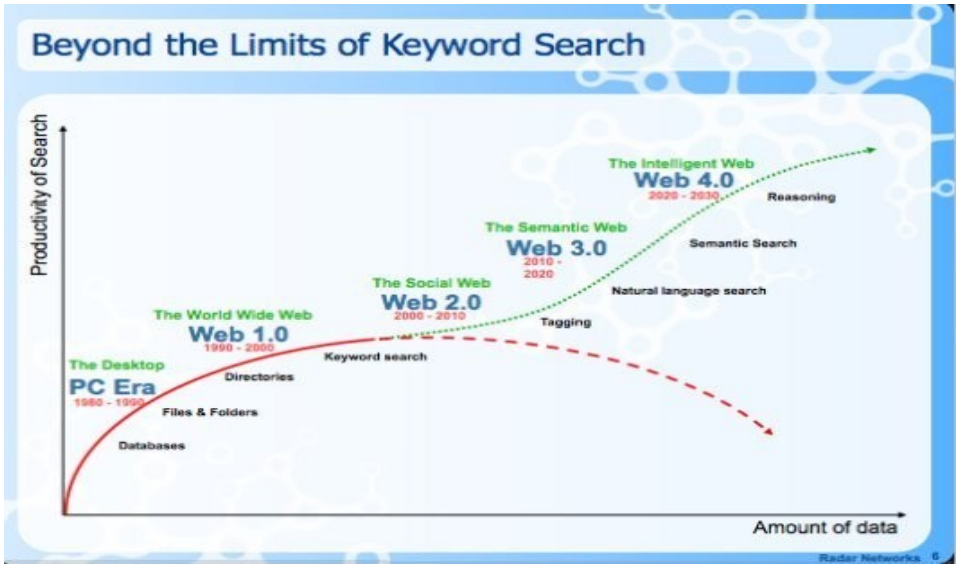
Istilah web semantik itu sendiri diperkenalkan oleh Tim Berners-Lee, penemu *World Wide Web*. Sekarang, prinsip web semantik disebut-sebut akan muncul pada Web 3.0, generasi ketiga dari *World Wide Web*. Bahkan Web 3.0 itu sendiri sering disamakan dengan web semantik. Web semantik menggunakan XML, XMLS (XML Schema), RDF, RDFS (*Resources Description Framework Schema*) dan OWL. Dimana seorang ahli yaitu Tim Berners Lee mengatakan "Orang-orang terus bertanya apa Web 3.0. Saya rasa mungkin ketika Anda punya *overlay scalable vector graphics* - semuanya beriak dan lipat dan berkabut mencari - pada Web 2.0 dan akses ke Web semantik terintegrasi di ruang besar data, Anda akan memiliki akses ke sebuah sumber data luar biasa ". [WIK10]. Berikut evolusi dari teknologi web dan perkembangannya :



Gambar II.2 . Teknologi Perkembangan Web – Secara Detail [AKR11]

Penjelasan :

Revolusi perkembangan teknologi web secara detail, dimulai dari era 90an yang sudah memasuki teknologi web 1.0. Di mulai dari web statis, lalu masuk ke dalam teknologi web 2.0 pada era 2010, sudah masuk dalam web dinamis yang memerlukan sebuah *database*. Sampai teknologi kedepanya web 3.0 dan web 4.0.



Gambar II.3. Teknologi Perkembangan Web – Secara Global [TUM11]

Penjelasan :

Gambar diatas merupakan evolusi secara global sebuah perkembangan web, dimulai dari web 1.0, web 2.0 hingga teknologi web 4.0. Pada dasarnya perkembangan itu menunjukan sebuah kemajuan yang cukup pesat. Tetapi, disamping itu dibalik pesatnya perkembangan web tersebut, maka kelemahan terhadap keamanan sistem inforamasi akan beragam pula pola dan perilakunya.

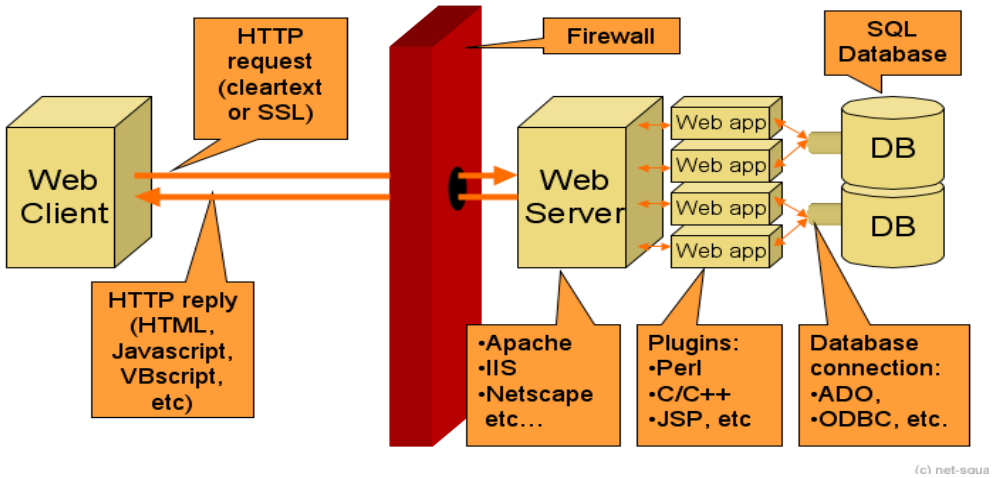
Kita bisa melihat betapa cepat teknologi perkembangan dari teknologi web tersebut. Seiring majunya teknologi web dari tahun ke tahun, begitu pula bermunculan ancaman (*threat*) pada keamanan informasi diiringi majunya teknologi web dan perkembangannya.

Kita bisa melihat semakin maju teknologi tersebut bukan berarti kita semakin nyaman dalam mengakses ataupun menggunakan teknologi tersebut, seperti halnya teknologi web 3.0 yang merupakan semantik web dimana web tersebut bukan hanya sebagai respons keinginan dari seorang user melainkan lebih dari sekedar itu, justru malah bisa mengerti keinginan manausia layaknya dalam berkomunikasi.

2.1.4. HTTP (*Hyper Text Transfer Protocol*)

Hyper Text Transfer Protocol (HTTP) merupakan sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia. Penggunaannya banyak pada pengambilan sumber daya yang saling terhubung dengan tautan, yang disebut dengan dokumen hiperteks, yang kemudian membentuk *World Wide Web* pada tahun 1990 oleh fisikawan Inggris, Tim Berners-Lee. Hingga kini, ada dua versi mayor dari protokol HTTP, yakni HTTP/1.0 yang menggunakan koneksi terpisah untuk setiap dokumen, dan HTTP/1.1 yang dapat menggunakan koneksi yang sama untuk melakukan transaksi. Dengan demikian, HTTP/1.1 bisa lebih cepat karena memang tidak usah membuang waktu untuk pembuatan koneksi berulang-ulang [WIK10].

HTTP ini merupakan standar internasional protokol jaringan yang sering kita gunakan ketika berselancar di dunia maya yang berjalan pada port 80, berikut komponen sistem aplikasi web, dapat digambarkan sebagai berikut :

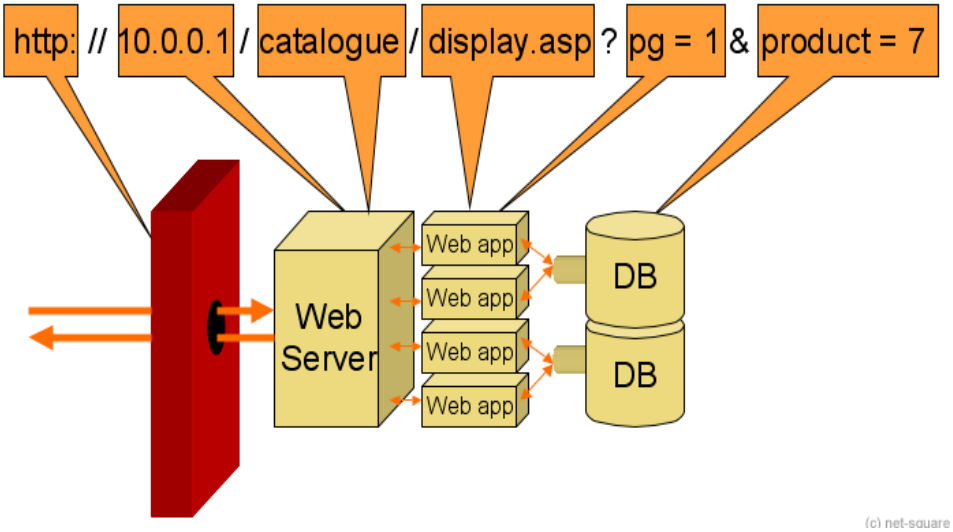


Gambar II.4. Cara Kerja HTTP [OBB10]

2.1.4.1. Komponen Web Aplikasi

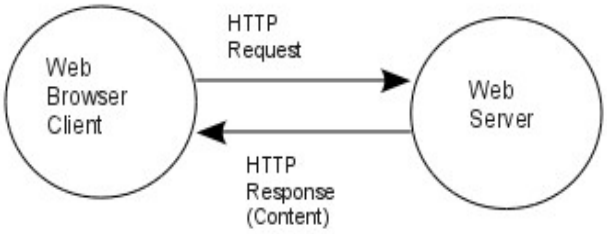
Komponen web aplikasi diatas merupakan secara global alur komunikasi antara user terhadap *web server* ketika terjadi permintaan atau *request* suatu

halaman web. *web server* yang berfungsi sebagai memberikan layanan terhadap client, jika seorang client meminta suatu halaman web maka *web server* tersebut akan proses ke *web application* jika client membutuhan *database* dari suatu web tersebut akan diteruskan ke databse server dan di kembalikan ke client dengan adanya permintaan *file* dari *database* tersebut, maka *website* menjadi dinamis. Berikut mapping cara kerja dari permohonan url ke dalam *web server*, sebagai berikut :



Gambar II.5. Komponen Web Aplikasi [JOSH10]

Web aplikasi yang menggunakan berbagai bahasa dengan lisensi *close source* maupun *open source* seperti halnya : *perl/c, php, asp, ruby, java* melayani *request logic business*, sehingga halaman web tersebut dapat ditampilkan ke client sesuai permohonan. *web server* yang melayani permohonan aplikasi tersebut seperti *apache web server*, *tomcat*, *microsoft IIS* dan *nginx*. Sehingga sebuah web aplikasi tidak bisa berjalan tanpa adanya *web server*, sebaliknya *web server* bisa berjalan tanpa web aplikasi. *Client* dan *Server* berkomunikasi dengan *Hypertext Transfer Protocol* dimana saat ini dengan versi terbaru yaitu HTTP/1.1, dimana alurnya kerja sistemnya seperti gambar berikut :



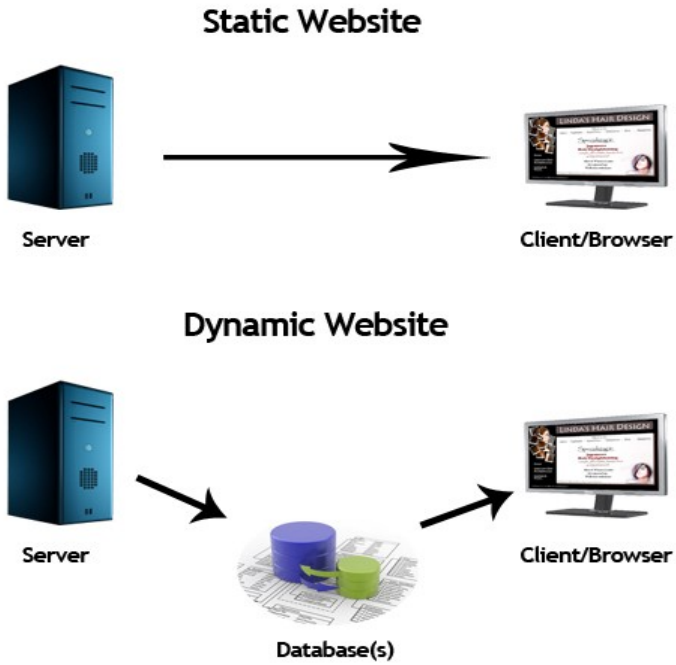
Gambar II.6. HTTP Request [MZS10]

Diagram diatas merupakan gambaran secara global permintaan dan pemberian *request* terhadap *client* dan *server* sehingga kita mengatahui secara dasar mekanisme dari komunikasi antara *client* dan *server*, dan memberikan pemahaman kita akan mekanisme tersebut secara dasar untuk membuat sebuah kebijakan keamanan informasi dalam web portal.

Beberapa metode dari *Hyper Text Transfer Protocol* (HTTP) pada komunikasi antara *client* dan *server*, diantaranya adalah :

- **POST** : layanan dari *web server* untuk memberikan *static* atau *dynamic content* terhadap *client*
- **GET** : layanan dari *web server* untuk memberikan *dynamic content* terhadap *client*
- **OPTION** : informasi untuk mendapatkan *file* atau *attribute*
- **HEAD** : sama dengan GET tetapi tidak ada data di *response body* atau badan program dari aplikasi tersebut
- **PUT** : perintah dari *client* untuk mengirim *file* ke *server*
- **DELETE** : perintah untuk menghapus *file* ke *server*
- **TRACE** : perintah echo untuk bisa di*response* oleh *body* pada *server*,

perintah ini berguna untuk *debugging* atau memeriksa kesalahan pada *server* tersebut. Metode *Hyper Text Transfer Protocol* dapat digambarkan sebagai berikut, sebagai contoh *reques*t terhadap *dynamic content* :



Gambar II.7. Website Statis dan Website Dinamis [SIT11]

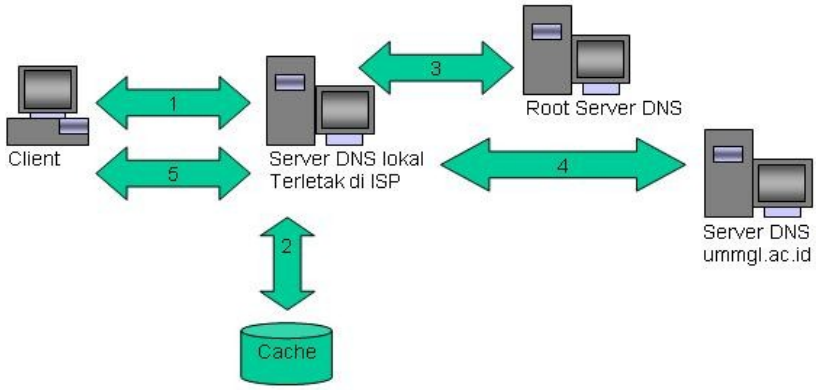
Gambar diatas merupakan sebuah ilustrasi dari sistem alur kerja pada web dinamis dan statis. Cukup jelas gambar diatas bahwa untuk web statis hanya berkomunikasi dengan *web server* itu sendiri, sedangkan web dinamis selain berkomunikasi dengan *web server*, juga meminta (*request*) dengan sebuah database.

Sebuah client meminta halaman konten dinamis menuju *web server* dengan perintah GET, setelah itu permintaan client di lanjutkan lagi ke dalam web aplikasi. Setelah dari web aplikasi diteruskan ke dalam *database* aplikasi tersebut. Kita bisa memahami cara kerja atau metode dari *Hyper Text Transfer Protokol*. Kita bisa membayankan ketika kita memberikan perintah seperti sintak *sql* atau *query database* agar bisa merubah konten web tersebut.

2.1.5. Domain Name Server (DNS)

DNS (Domain Name System) adalah sebuah sistem yang menyimpan informasi tentang nama *host* maupun nama domain dalam bentuk basis data

tersebar (*distributed database*) di dalam jaringan komputer, misalkan: Internet. DNS menyediakan alamat IP untuk setiap nama host dan mendata setiap *mail exchange server* yang menerima surat elektronik (*email*) untuk setiap domain. DNS menyediakan servis yang cukup penting untuk Internet, bilamana perangkat keras komputer dan jaringan bekerja dengan alamat IP untuk mengerjakan tugas seperti pengalamatan dan penjaluran (*routing*), manusia pada umumnya lebih memilih untuk menggunakan nama host dan nama domain, contohnya adalah penunjukan sumber *Uniform Resources Locator* (URL) dan alamat e-mail. DNS menghubungkan kebutuhan ini [WIK10]. Berikut ilustrasi dari cara kerja suatu D*omain Name Server* ketika kita me*request* ke suatu halaman *website* di dalam maupun luar. Perlu kita sadari bahwa munculnya teknolog baru bukan berarti membawa suatu kemajuan yang cukup lebih berarti, oleh karenanya tedapat analisa atau audit terhadap teknologi yang kita adopsi.



Gambar II.8. Cara Kerja DNS [CON11]

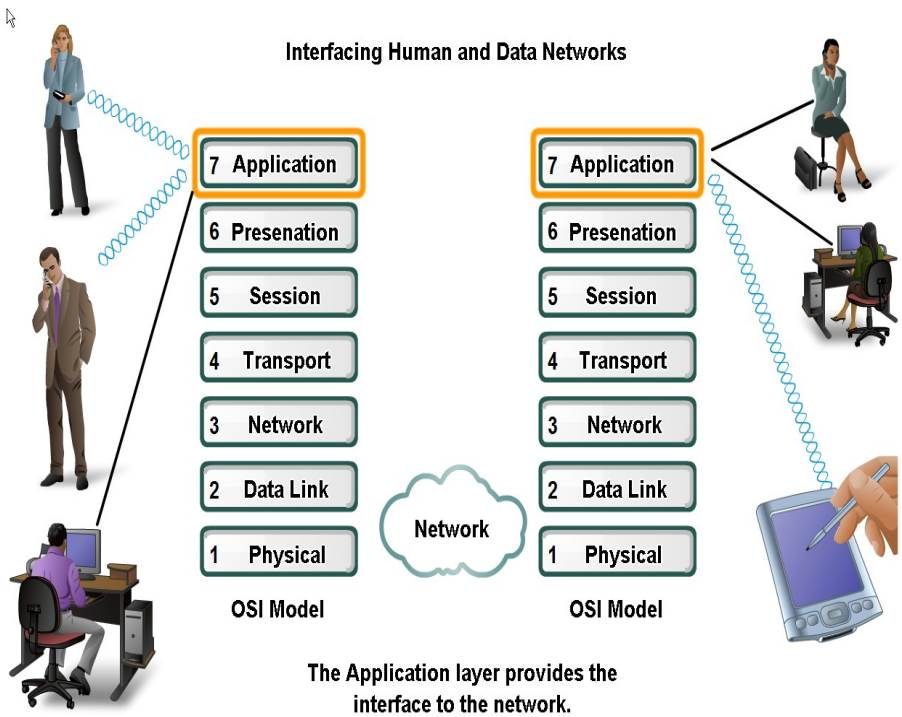
Gambar diatas menjelaskan tentang alur kerja sebuah DNS *server*, dimana sebuah *client request* domain http://www.ummgl.ac.id melalui *cache* pada yang terletak pada *server* DNS lokal, jika *cache* tersebut tersedia maka di kirim ke *client* jika tidak maka akan menuju sebuah *root server* DNS jika tidak tersedia

maka akan di *respons* ke *server* DNS ummgl.ac.id dan begitu seterusnya.

Gambar II.9. Jenis Nama Domain [SEN10]

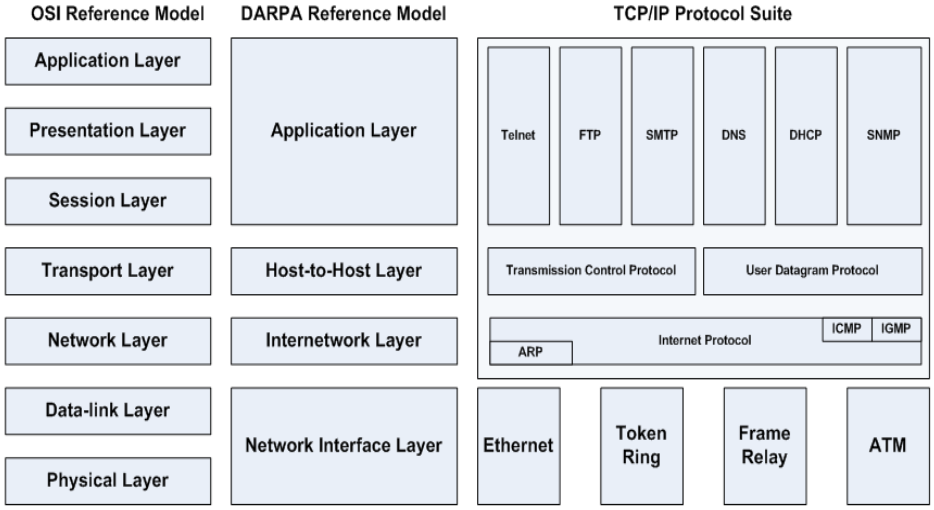
2.1.6. Model OSI (*Open System Interconnection*) Layer

OSI Reference Model merupakan sebuah model ideal dari koneksi logis yang harus terjadi agar komunikasi data dalam jaringan dapat berlangsung. Beberapa protokol yang digunakan dalam dunia nyata, semacam TCP/IP, DECnet dan IBM *Systems Network Architecture* (SNA) memetakan tumpukan protokol (*protocol stack*) mereka ke *OSI Reference Model*. *OSI Reference Model* pun digunakan sebagai titik awal untuk mempelajari bagaimana beberapa protokol jaringan di dalam sebuah kumpulan protokol dapat berfungsi dan berinteraksi [WIK10] , seperti ilustrasi gambar berikut :



Gambar II.10. OSI Application Layer [SAA10]

Gambar diatas mengilustrasikan kumpulan protokol yang saling berinteraksi pada layer aplikasi, begitu pula hubungan sebuah koneksi jaringan yang terhubung internet menggunakan media web browser. Jika kita lebih rinci lagi untuk masing-masing layer itu mempunya beberapa tugas dalam pembagian layer, sebagai berikut :



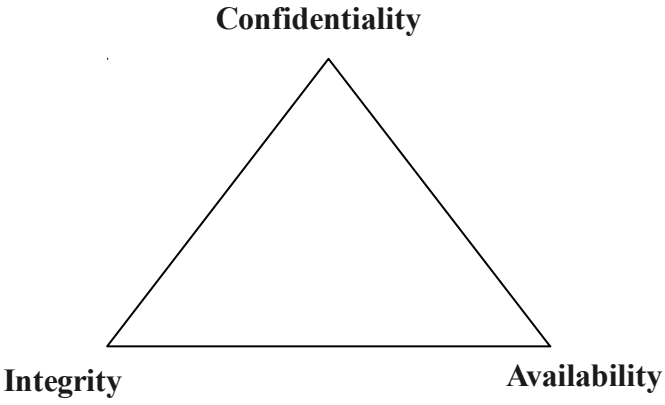
Gambar II.11. OSI Layer [WP10]

Menurut saya ini merupakan pemahaman awal ketika kita akan mempelajari keamanan informasi. Karena tujuh layer ini merupakan standart model utama dalam komunikasi dari satu *host* ke *host* yang lain. Seperti halnya layer aplikasi,merupakan layer utama dalam komunikasi dimana dapat dituangkan dalam implementasi aplikasi yang kita gunakan sehari-hari .

2.1.7. Parameter Keamanan Sistem Informasi

Dalam melakukan penilaian terhadap tingkat keamanan sebuah sistem informasi (SI), kita harus memiliki sebuah acuan, agar sistem tersebut dapat dikategorikan sebagai sistem yang aman. Krutz dan Vinez [KRU03] menjelaskan bahwa keamanan dari sebuah SI adalah segala macam cara yang bertujuan untuk menjaga faktor *Confidentiality*, *Integrity* dan *Availability* dari SI tersebut. Oleh

karenanya itu, sebuah SI dapat dikatakan aman jika ketiga faktor tersebut telah dipenuhi.



Gambar II.12. Konsep Security

Bishop [BIS04] menjelaskan secara mendetail tentang ketiga faktor tersebut :

a. *Confidentiality* (Kerahasiaan)

Confidentiality memiliki arti penyembunyian informasi atau sumber daya. Penyembunyian disini berarti bahwa informasi atau sumber daya tersebut hanya dapat diakses oleh pihak yang berhak, dengan kata lain "tersembunyi" dari pihak-pihak yang tidak memiliki hak terhadapnya. Kebutuhan untuk menjaga informasi tetap rahasia bermula dari penggunaan komputer pada organisasi yang sensitif seperti pemerintahan dan industri. Dengan kata lain sebuah informasi yang disampaikan harus sesuai

dengan orang yang berhak menerima informasi tersebut.

b. *Integrity* (Kesatuan)

Sebuah keadaan dimana suatu entitas (dalam hal ini SI) dapat dipercaya, yang berarti tidak dapat dirubah isi dari informasi tersebut. Dalam hal ini *integrity* harus mempunyai sebuah tanggungjawab pencegahan terhadap modifikasi atas informasi yang disampaikan dan menjaga agar informasi tersebut tetap konsisten adanya.

c. ***Availability*** **(Ketersediaan)**

Availability memiliki arti ketersediaan sumber daya dan informasiyang ada ketika dibutuhkan. Faktor dari *availability* yang terkait adalah sebuah sistem dimana layanan itu tidak boleh berhenti hingga pada saat harus menghentikan layanannya dari hak yang dapat di percaya, dengan kata lain semua data dan informasi yang sifatnya layanan publik harus selalu tersedia dan siap setiap saat untuk bisa di akses. Seperti halnya sebuah layanan informasi yang di publish dalam web site portal tidak harus selalu tersedia konten dan layanannya dan tidak boleh lumpuh sama sekali.

Pentingnya mengamankan informasi merupakan upaya wajib untuk menjaga keutuhan, validitas, dan ketersediaan dari informasi tersebut. Kita ketahui bahwasanya jaman era teknologi informasi ini saling terhubung satu sama lain sehingga keterbukaan sistem pada suatu infomasi yang dimiliki oleh orang lain amat sangat mungkin dapat diakses dengan cara mudah dan singkat jika kita tidak menjaga informasi tersebut. Apalagi lahirnya perusahaan *search engine* terbesar di dunia seperti *google.com*, maka semakin mudah seseorang untuk memperoleh informasi mengunakan *search engine* yang dimilikinya. Informasi adalah salah satu aset bagi sebuah perusahaan atau organisasi, yang sebagaimana aset lainnya memiliki nilai tertentu bagi perusahaan atau organisasi tersebut sehingga harus dilindungi, untuk menjamin kelangsungan perusahaan atau organisasi, meminimalisir kerusakan karena kebocoran sistem keamanan informasi, mempercepat kembalinya investasi dan memperluas peluang usaha.

2.1.7.1. Kontrol Terhadap Keamanan Informasi

Keamanan SI yang diadopsi oleh sebuah organisasi haruslah di kontrol, hal ini bertujuan untuk mengurangi efek dari ancaman serta kerentanan terhadap keamanan SI, sehingga mencapai tingkat yang bisa diterima oleh organisasi [KRU01]. Proses kontrol ini adalah dengan menentukan akibat dari ancaman yan mungkin berakibat terhadap organisasi dan kemungkinan – kemungkinan dimana ancaman tersebut menjadi kenyataan. Dengan melakukan kontrolini, maka akibat

yang ditimbulkan dapat ditekan sampai kepada level yang bisa diterima menurut kebijakan organisasi.

2.1.7.2. Penilaian Terhadap Ancaman dan Kerentanan Sistem Informasi

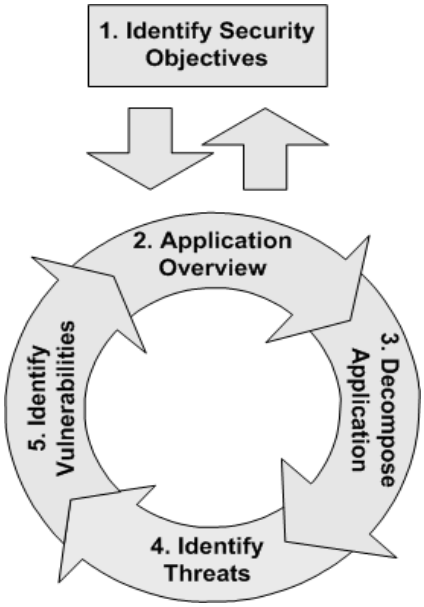
Setelah penilaian aset yang ada serta mempunyai nilai yang cukup besar, langkah selanjutnya dalam manajemen resiko SI adalah melakukan penilaian terhadap ancaman dan kerentanan yang melekat pada Sistem Informasi tersebut. Penilaian disini terkait dengan klasifikasi tentang apa saja yang dikategorikan sebagai ancaman, kemungkinan ancaman tersebut terjadi. Meskipun beberapa resiko dapat mempengaruhi semua aset, namun beberapa resiko yang spesifik saja yang berpengaruh terhadap resiko tersebut . Adapun sebuah rumus resiko tersebut :

Risk = Threat + Vulnerbility + Asset Value

Resiko merupakan sebuah ancaman dari lingkungan ekternal maupun internal ditambah kelemahan dari sistem tersebut serta ditambah pula nilai asset yang ada pada sistem tersebut, namun resiko itu bisa diminimalisasi dengan cara mencermati apa saja ancaman yang terdapat atau yang timbul jika ada kelemahan dari sistem kita dan seberapa besar resiko untuk nilai asset yang akan hilang akibat dari ancaman tersebut jika kita memilih teknologi itu [PEN11].

2.1.8. Threat Risk Modeling

Threat Risk Modeling merupakan suatu tahapan proses untuk membangun aplikasi, dimana memiliki 5 tahapan untuk melalui proses tersebut, diantaranya adalah sebagai berikut jika digambarkan dalam bagan atau alur *flowchart*nya :



Gambar II.13 . Threat Risk Modeling
[OWS11]

Kelima tahapan diatas sebagai berikut :

- ***Identify Security Objectives***, disini hal yang paling utama ketika akan membangun suatu aplikasi agar aman adalah mengidentifikasi objek keamanan dari aplikasi tersebut. Ruang lingkup atau scope aplikasi itu tipenya seperti apa ? Jika berbentuk database , maka asset utama yang paling objektif untuk diamankan adalah aplikasi database tersebut.
- ***Application Overview***, melihat secara keseluruhan aplikasi dari karakter dan ancaman yang menjadi resiko terjadinya serangan.
- ***Decompose Application***, merinci secara detail dari aplikasi tersebut untuk lebih detail dari spesifikasi aplkasi tersebut serta ancaman aplikasi yang akan di bangun.
- ***Identify Threat***, setelah melewati tahapan 2 dan 3 tersebut yaitu mengidentifikasi ancaman dari konteks dan skenario aplikasi yang akan di bangun itu.

- ***Identify Vulnerbility***, *review* seluruh lapisan aplikasi untuk mengidentifikasi kelemahan dari ancaman serta gunakan kategori untuk fokus kedalam area yang dijadikan sebagai objek keaman informasi.

Model diatas amat sangat membantu dalam membuat atau merumuskan model ancaman pada suatu aplikasi yang akan kita bangun serta meminilisasi resiko terjadinya serangan-serangan yang bisa mematahkan sistem aplikasi tersebut. Adapun proses dari *threat modeling* diatas adalah :

Gambar II.14. Process Threat Modeling [CIP11]

Beberapa tahapan proses dari threat modeling dalam keamanan web aplikasi, sebagai berikut :

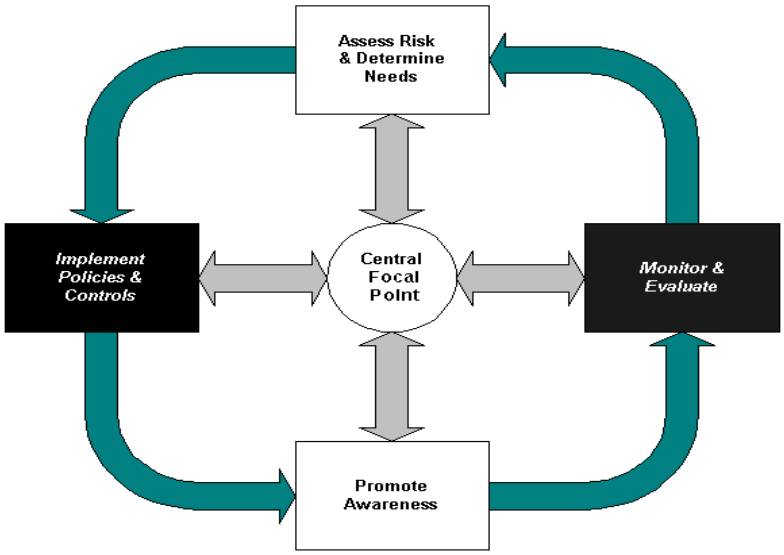
- ***Security Requirement***, kebutuhan akan suatu keamanan, merupakan podasi utama dalam membangun sebuah keamanaan informasi agar mencegah terjadinya salah desain model suatu keamanan infomasi. Kebutuhan disini adalah ruang lingkup atau *scope* dari keamanan yang ada pada sistem tersebut.
- ***Asset Identification***, identifikasi asset pada suatu sistem yang akan diamankan dari ancaman – ancaman yang ada. Dengan adanya identifikasi aset tersebut maka akan terjadi minimalisasi dari resiko yang akan terjadi jika terjadi

serangan pada sistem tersebut.

- ***Access Matrix,*** akses disini adalah jangkauan aktivitas pada serangan-serangan atau yang membuka akses masuk ke dalam sistem tersebut.
- ***Threat Identification,*** ancaman – ancaman yang ada atau serangan – serangan yang terjadi di sistem tersebut harus diidentifikasi pola atau *pattern* tingkah laku serangan tersebut untuk bisa dipelajari dan solusinya
- ***Simulate Attack Scenario,*** setelah kita pelajari serangan – serangan yang ada, maka simulasi skenario serangan harus sudah terbentuk dan dapat dipahami secara logika dari pola serangan yang ada. Sehingga kita bisa membuat suatu *model prevention* dari serangan tersebut
- ***Test Attack Condition*****,** setelah simulasi serangan tersebut berhasil, maka kita bisa melakukan testing serangan dengan kondisi yang ada. Guna untuk mengetahui dan mengukur sejauh mana keamanan pada sistem yang telah dibangun tersebut
- ***Reporting,*** tahap alkhir adalah laporan hasil rumusan dari serangan – serangan yang ada pada sistem tersebut sebagai dokumentasi pada entitas terkait.

2.1.8.1. Daur Hidup Manajemen Resiko (*The Risk Cycle Management*)

Pada dasarnya semua semua tehnik atau metode tidak ada yang nyaman ketika kita implementasikan dan semua mempunyai resiko yang harus dihadapi pada masing-masing entitas. Tetapi disni ada sebuah metode untuk memanage resiko yang ada dengan cara meminimalisasi resiko yang akan terjadi. Berikut dibawah ini merupakan skema dari daur hidup manajemen resiko.



Gambar II.15. The Risk Management Cycle [OWS11]

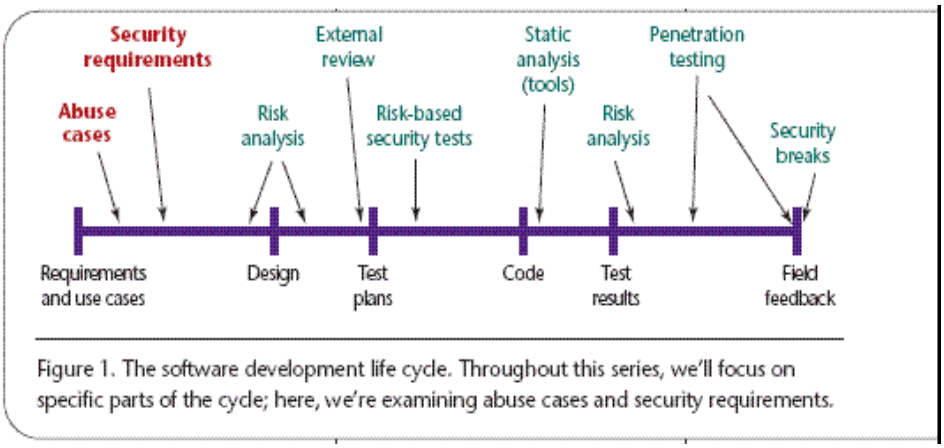
Kita bisa lihat diatas, ada empat (4) titik atau node yang dapat mengatur resiko dan meminimalisasikannya dalam sistem. Yaitu sebagai berikut :

1. ***Assesment Risk and Determine Needs***, disini kita harus menaksir yang mungkin akan terjadi dan mendeteminasikan keperluan resiko tersebut.
2. ***Implement Polices and Control*****,** mengimplementasikan kebijakan dan kontrol untuk bisa menjalankan masing-masing sistem untuk tujuan yang sama.
3. ***Promote Awareness****,* mempromposikan kesadaran, cek point kesadaran disini adalah pentingnya menjalankan budaya saling mengingatkan terhadap sistem yang sudah dijalannkan untuk tidak keluar dari sistem tersebut.
4. ***Monitor and Evaluation****,* tahapan akhirnya adalah selalu memonitor, mengawasi serta mengevaluasi sistem yang sudah mulai usang dan memicu resiko berkepanjangan

Keempat tahapan diatas merupakan tolak ukur dasar dalam mengatur dan meminimalisasi resiko yang mungkin terjadi sangat fatal, oleh karenanya kita harus menjaga 4 entitas tersebut jangan sampai keluar dari jalurnya

2.1.8.2. Secure Development Lifecycle (SDL)

Secure Development Lifecycle (SDL) merupakan daur hidup membangun suatu aplikasi dengan menitik beratkan pada keamanan infomasi. Ada beberapa tahapan dari SDL ini, antara lain adalah :



Gambar II.16. Secure Development Life Cycle [TSA08]

Setelah melalui tahapan diatas bisa dikatakan bahwa aplikasi yang kita bangun dari awal sampai akhir menjadi aplikasi yang memenuhi standar keamanan yang cukup baik. Entah itu dari sisi desain, *development* hingga *testing* dan *review* dan menjadi suatu produk yang aman.Alur dari prinsip diatas adalah kita melakukan ada beberapa tahap antaranya adalah melihat kebutuhan dari keamanan sistem tersebut atau aplikasi yang akan dibangun, setelah itu pada tahap desain dipertimbangkan juga resiko yang akan didapat atas desain yang sudah dibuat. Setelah itu melakukan sebuah *development code* untuk membangun aplikasi tersebut dan melakukan *penetration testing* , tidak luput juga harus membuat hasil laporan dari *penetration testing* untuk evaluasi dari sistem yang sudah dibangun.

2.1.8.3. Kerangka Kerja Keamanan (Security Framework) / SD^3+C

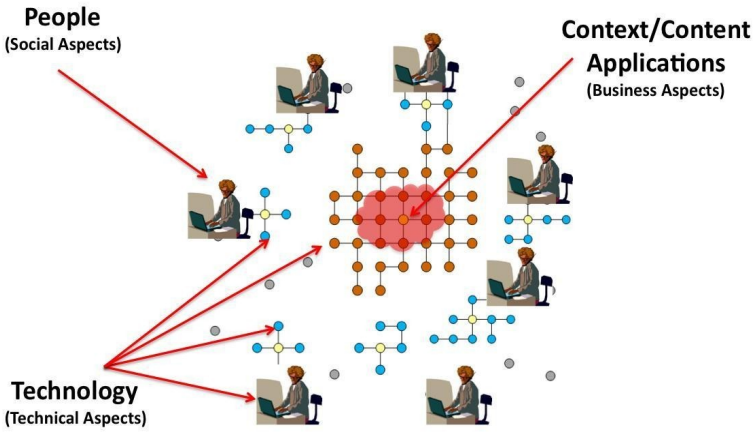
Beberapa kriteria dan parimeter suatu keamanan pada sistem yang menjadi tolak ukur suatu pekerjaan serta pengertian yang akan kita implementasikan salah satunya adalah sebagai gambar berikut ini :

Gambar II.17. Framework SD^3 + C [RG11]

Empat point diatas yaitu : *secure by design, secure by default, secure by deployment* dan *communication* merupakan penjabaran 4 point / kriteria dalam *development* (membangun) suatu aplikasi. Inti dari framework diatas merupakan proses dari macam kebutuhan dalam keamanan aplikasi dan kebutuhannya. Yang terpenting adalah bagaimana kita mengimplemtasikan dari beberapa item tersebut sehingga *output* aplikasi yang telah di develop menjadi sesuai keinginan kita.

2.1.8.4. Rantai Jaringan Internet

Rantai jaringan internet merupakan hubungan dari ketiga entitas dalam pengamanan informasi dalam konteks keamanan. Oleh karenanya ketiga entitas itu saling terhubung satu sama lain, seperti gambar berikut ini :



Gambar II.18. Rantai Jaringan Internet [REI10]

Penjelasan :

1. Internet merupakan suatu jejaring raksasa yang mempertemukan berbagai jaringan komputer yang ada di muka bumi ini. Jejaring raksasa tersebut terjadi dengan cara menghubungkan beraneka ragam pusat-pusat penyimpanan data dan informasi yang tersebar lokasinya di seluruh dunia dengan memanfaatkan teknologi informasi dan komunikasi. Berbagai peralatan teknis dan piranti teknologi ini merupakan kunci terciptanya sebuah jaringan raksasa yang tumbuh secara eksponensial dari waktu ke waktu.

2. Jejaring raksasa ini pada dasarnya merupakan sebuah infrastruktur komunikasi yang di atasnya dapat diimplementasikan berbagai aplikasi untuk memenuhi sejumlah kebutuhan hidup manusia, seperti: keperluan pendidikan, interaksi sosial, transaksi bisnis, pengembangan pribadi, dan lain sebagainya. Konteks pertukaran barang dan jasa tersebut (baca: bisnis) kerap mendominasi pemanfaatan infrastruktur internet ini.

3. Pada akhirnya, sang pengguna berbagai aplikasi yang berjalan di atas internet ini adalah para individu atau komunitas yang berkepentingan, mulai dari anak-

anak, orang tua, karyawan, pengusaha, seniman, politikus, pendidik, wiraswastawan, dan lain sebagainya. Melihat adanya keterhubungan yang jelas baik secara fisik maupun virtual antara ketiga komponen ini, maka dapat disimpulkan bahwa kunci sukses tidaknya atau tinggi rendahnya tingkat keamanan internet sangat ditentukan oleh setiap perangkat teknis, setiap aplikasi, dan setiap individu yang mempergunakannya [REI10]

Ketiga entitas itu merupakan gambaran aktifitas atau rangkaian yang saling terhubung antara pengguna internet yaitu People, sedangkan Teknologi sebagai fasilitas penghubung jaringan yang terhubung satu dengan lainnya dan aplikasi sebagai media komunikasi untuk melakukan sebuah kegiatan masing-masing individu. Rantai jaringan diatas merupakan sebuah gambaran sistem hubungan dalam dunia virtual yang melibatkan antara *People, Technology, Application,* dimana ketiga aspek tersebut menentukan proses dari sebuah keamanan informasi.

2.2. Insiden yang Terjadi Pada Dunia Maya

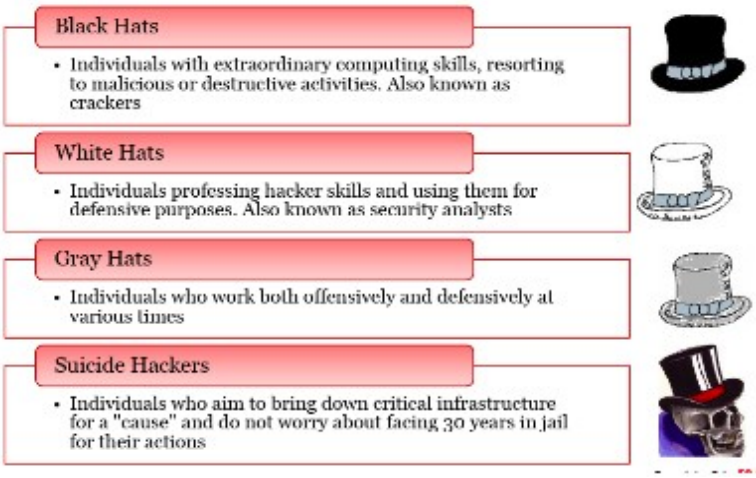
Penulis melakukan beberapa tinjauan studi pada beberapa website yang telah melakukan riset yang terkait pada penelitian tentang keamanan web site portal. Guna untuk menamba referensi dan pengetahuan tentang tema penelitian tersebut, sehingga kualitas dari penelitian ini menjadi baik. Di bawah ini merupakan sebuat tabel kegiatan yang berkenaan dengang keamanan informasi yang dapat di perjual belikan dalam nilai dollar [REI10].

Item	Advertised Price (in US Dollars)
United States-based credit card with card verification value	$1–$6
United Kingdom-based credit card with card verification value	$2–$12
An identity (including US bank account, credit card, date of birth, and government issued identification number)	$14–$18
List of 29,000 emails	$5
Online banking account with a $9,900 balance	$300
Yahoo Mail cookie exploit—advertised to facilitate full access when successful	$3
Valid Yahoo and Hotmail email cookies	$3
Compromised computer	$6–$20
Phishing Web site hosting—per site	$3–5
Verified PayPal account with balance (balance varies)	$50–$500
Unverified PayPal account with balance (balance varies)	$10–$50
Skype account	$12
World of Warcraft account—one month duration	$10

Gambar II.19. Underground Economy Threat [REI10]

Kita bisa lihat betapa mudahnya informasi dapat di beli dengan nilai dollar yang tidak cukup besar dan kita bisa mendapatkan informasi sesuai keinginan kita. Kegiatan kriminal pun dapat diperjualbelikan dalam dunia maya, gambaran tersebut bisa kita ilutrasikan betapa tidak amannya dunia maya terhadap keamanan informasi saat ini.

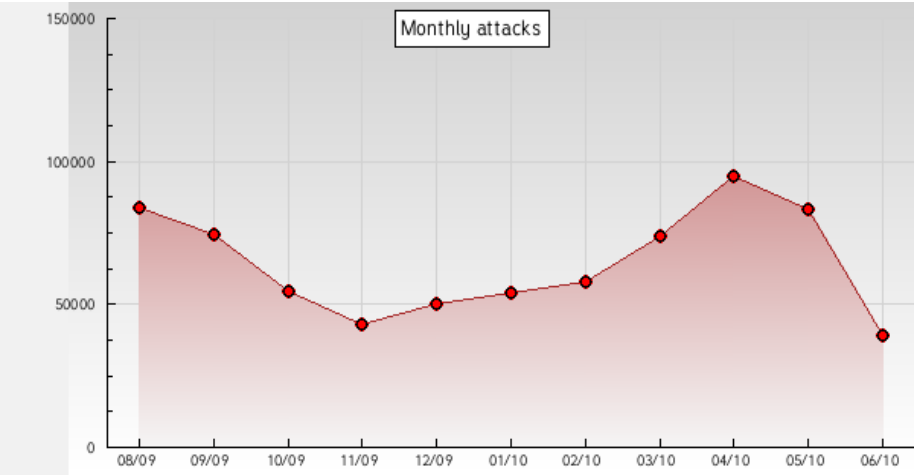
Pelakunya pun dapat digolongkan sebagai berikut sebagai ancaman dalam dunia maya, dan kita bisa menilainya :



Gambar II.20. Hacker Threat [REI10]

Berikut informasi serangan-serangan *web defacement* dari sebuah situs yang melaporkan secara keseluruhan insiden-insiden serangan web portal dalam satu

bulannya :



Gambar II.21. Statistik Bulanan Tahun 2010 Serangan Defacement [ZON10]

Berikut statistik metode serangan (*attack method*) yang sering terjadi dari beberapa situs web portal di dalam dunia maya :

Dan berbagai alasan seorang hacker untuk melakukan insiden tersebut, berikut informasinya :

Attack Reason	Year 2008	Year 2009	Year 2010
I just want to be the best defacer	201.270	122.442	78.761
Heh just for fun!	96.438	176.725	179.707
As a challenge	61.112	26.921	13.422
Political reasons	50.578	72.767	19.360
Patriotism	46.619	40.374	17.877
Revenge against that website	4.802	23.513	15.147
Not available	56.640	81.667	28.545

Gambar II.23. Attack Reason [ZON10]

Sangat ironis sekali ketika kita teliti berbagai statistik diatas yang mempunyai alasan dan metode yang berbeda-beda. Lalu bagaimana keamanan informasi di indonesia yang khususnya harus melindungi web portal pemerintah

sebelum terjadi. Apa rencana pemerintah selanjutnya ? Apakah web portal pemerintah indonesia khususnya masuk kedalam statistik diatas.

Berikut perbandingan serangan pada sistem informasi linux dan windows dalam statistik berikut :

Year	Total defacements Linux (all distros)	Total defacements Windows (all version
2000	931	2.587
2001	4.080	13.549
2002	22.693	43.441
2003	191.720	58.571
2004	247.113	119.402
2005	276.294	179.945
2006	446.039	258.129
2007	305.968	139.427
2008	352.449	141.061
2009	378.728	143.151
2010	256.648	87.959
Total	**2.482,663**	**1.187,222**

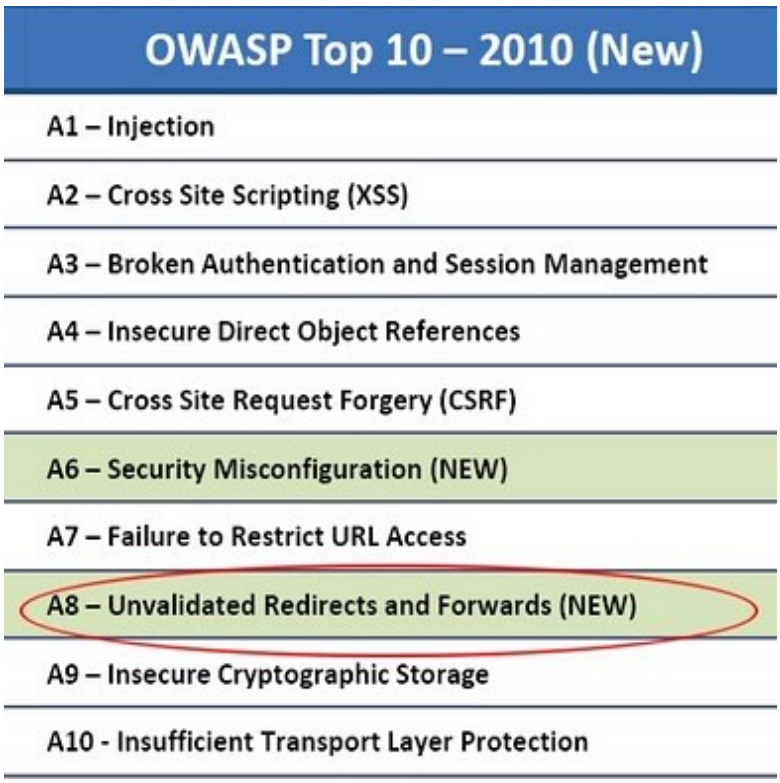
Gambar II.24. Total Defacement 2010 [ZON11]

Statistik diatas menunjukan bahwa sistem operasi linux lebih banyak di banding sistem operasi windows. Ini menunjukan bahwasanya kedua sistem operasi mempunyai kelemahan dan kekurangannya dimana bisa disebabkan masalah *misconfoguration* atau kelalaian seorang administrator untuk disiplin dalam *patching system.* Pada dasarnya memang dalam dunia ini tidak ada yang kekal begitu juga sebuah sistem teknologi informasi secanggih teknologi apapun tidak ada yang *secure* atau aman bahwasanaya ***security is process*** dan ***bukan teknologi.***

2.2.1. *Open Web Application Security Project* (OWASP)

Open Web Application Security Project (OWASP) merupakan sebuah

komunitas keamanan web aplikasi yang membuat serbuah project dan riset tentang keamanan informasi khususnya web aplikasi. Berdasarkan statistik dari komunitas tersebut ada 10 ancaman keamanan web aplikasi yang sedang tren saat ini khususnya tahun 2010 , sebagai berikut :



Gambar II.25. OWASP Top 10 – 2010 [OWS11]

Penjelasan :

A1- *Injection*

Kelemahan injeksi, seperti injeksi SQL, OS, dan LDAP, terjadi ketika data yang tidak dapat dipercaya dikirim ke suatu interpreter sebagai bagian dari suatu perintah atau query. Data berbahaya dari penyerang tersebut dapat mengelabui interpreter untuk mengeksekusi perintah yang tidak direncanakan, atau untuk mengakses data yang tidak terotorisasi.

A2 - *Cross Site Scripting (XSS)*

Kelemahan XSS terjadi ketika aplikasi mengambil data yang tidak dapat

dipercaya dan mengirimnya ke suatu web browser tanpa validasi yang memadai. XSS memungkinkan penyerang mengeksekusi script-script di dalam browser korban, yang dapat membajak sesi pengguna, mengubah tampilan website, atau mengarahkan pengguna ke situs-situs jahat.

A3 – *Broken Authentication and Session Management*

Fungsi-fungsi aplikasi yang berhubungan dengan otentikasi dan pengelolaan sesi seringkali tidak dimplementasikan dengan benar. Hal ini memungkinkan penyerang mendapatkan password, key, Pengelolaan Sesi dan token-token sesi, atau mengeksploitasi cacat implementasi lainnya untuk memperoleh identitas yang buruk pengguna yang lain .

A4 - *Insecure Direct Object Management*

Direct object reference terjadi ketika pengembang mengekspos referensi ke suatu objek implementasi internal, seperti file, direktori, atau kunci database. Tanpa adanya suatu pemeriksaan kendali akses atau perlindungan lainnya, penyerang dapat memanipulasi referensi-referensi ini untuk mengakses data yang tidak terotorisasi.

A5 - *Cross Site Request Foregery (CSRF)*

Suatu serangan CSRF memaksa browser korban yang sudah log-on untuk mengirim HTTP request yang dipalsukan, termasuk di dalamnya session cookie korban dan informasi otentikasi lain yang otomatis disertakan, ke suatu aplikasi web yang rentan. Hal ini memungkinkan penyerang untuk memaksa browser korban menghasilkan request yang dianggap sah oleh aplikasi rentan tadi.

A6 - *Security Miss Configuration*

Keamanan yang baik mensyaratkan dimilikinya suatu konfigurasi keamanan (yang terdefinisi dan diterapkan) untuk aplikasi, framework, server aplikasi, *web server*, server database, dan platform. Semua pengaturan ini harus didefinisikan, diimplementasikan,dan dipelihara, karena terdapat banyak aplikasi yang dirilis tanpa konfigurasi default yang aman. Hal ini juga mencakup menjaga semua software up-to-date, termasuk semua pustaka kode yang digunakan aplikasi tersebut.

A7 - *Failure Restric URL Access*

Banyak aplikasi web yang tidak melindungi data sensitif (seperti data kartu kredit, SSN, kredensial otentikasi) dengan enkripsi atau hashing yang memadai. Penyerang dapat mencuri atau memodifikasi data dengan perlindungan lemah semacam itu untuk melakukan pencurian identitas, kejahatan kartu kredit, atau kriminalitas lain.

A8 - *Unvalidated Redirects and Forward (New)*

Banyak aplikasi web memeriksa hak akses URL sebelum memberikan link dan tombol-tombol yang diproteksi. Bagaimanapun juga, aplikasi perlu melakukan pemeriksaan kendali akses yang serupa setiap kali halaman-halaman ini diakses, atau penyerang akan dapat memalsukan URL untuk mengakses halaman-halaman yang tersembunyi ini

A9 - *Insecure Cryptographic Storages*

Aplikasi seringkali gagal untuk mengotentikasi, mengenkripsi, dan melindungi kerahasiaan serta integritas lalu-lintas jaringan yang sensitif. Ketika aplikasi gagal melakukan hal-hal tersebut, adalah dikarenakan ia mendukung algoritma yang lemah, menggunakan sertifikat yang tidak valid atau sudah kadaluarsa, atau karena tidak menggunakannya dengan benar .

A10 - *Insufficient Transport Layer Protection*

Aplikasi web seringkali mengarahkan (redirect) dan meneruskan (forward) pengguna ke halaman data yang tidak menentukan halaman Forward yang Tidak dan website lain, dan mengunakan penyerang dapat dapat dipercaya untuk ke situs phishing atau tujuan. Tanpa validasi yang tepat, mengarahkan korban divalidasi malware, atau menggunakan forward untuk mengakses halaman yang tidak terotorisasi. Diatas merupakan serangan serta ancaman yang masuk dalam top 10 tahun 2010 bagi keamanan web aplikasi terutama web teknologi 2.0.

2.2.2. White Hat Security

White Hat Security merupakan lembaga keamanan informasi *comercial* yang juga konsentrasi terhadap riset keamanan informasi khususnya web site portal di

negara Amerika Serikat. Lembaga ini mengeluarkan statistik ancaman pada keamanan web aplikasi khususnya web site portal, sebagai berikut [WHS10] :

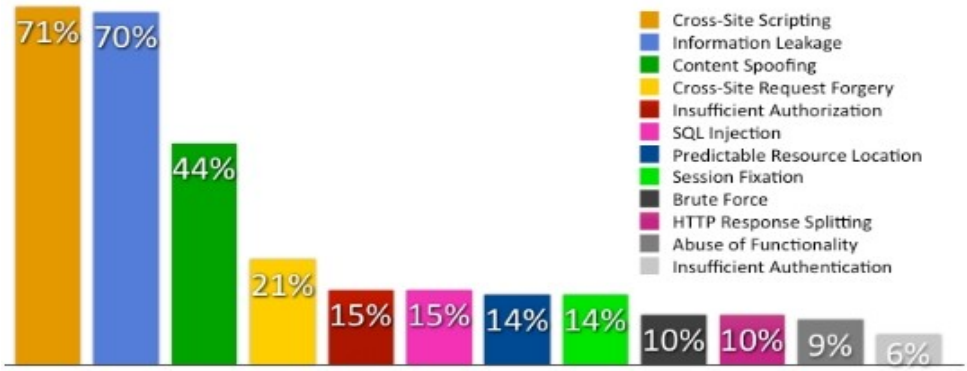


Figure 5. Overall Top Vulnerability Classes
Sorted by Percentage Likelihood

Gambar II.26 Threat Web Application 2010 [WHS10]

2.2.3. E-Government Security Risk Management

Dengan populernya dan majunya teknologi jaringan komputer, berbagi secara *real-time* dalam jumlah besar di sebuah instansi pemerintah dan pertukaran informasi dua arah sudah dimungkinkan teknis. E-Pemerintah adalah sebuah bidang administrasi pemerintah yang didasarkan pada teknologi informasi. Inti dari e-pemerintah adalah menggunakan teknologi informasi elektronik untuk memecahkan batas administrasi organisasi, dan membangun sebuah virtual elektronik pemerintah. Orang bisa mendapatkan informasi dan pelayanan pemerintah melalui media elektronik. Pemerintah dapat berkomunikasi dengan satu sama lain melalui berbagai jenis media elektronik yang dapat digunakan di dalam badan-badan pemerintah, antara pemerintah yang berbeda, atau antara pemerintah dan masyarakat.

Ada tiga prosedur dalam manajemen resiko, terkait dalam e-pemerintah, antara lain adalah :

- **Risk Identifying**, identifikasi risiko didasarkan pada pengumpulan

berbagai ancaman yang relevan terhadap sistem *e-government*, *bug* dan penanganan yang sesuai, dan kemudian menyimpulkan kemungkinan resiko atau potensi ancaman ke sistem *e-governmen*t.

- **Risk Analyzing,** ancaman sumber dapat menjadi semua jenis lingkungan atau peristiwa termasuk orang-orang, alam, dan sebagainya, yang merugikan terhadap sistem. Ancaman alami yang dihadapi berkaitan dengan sistem yang lokasi geografis, namun ancaman dari orang-orang mungkin tidak memiliki maksud atau tujuan. Untuk mengidentifikasi ancaman sistem yang dihadapi, kita dapat menggunakan metode yang berbeda, seperti brainstorming, analisis skenario.
- **Risk Controlling**, memilih dan menggunakan beberapa pengendalian risiko metode untuk menjamin risiko dapat dikurangi ke tingkat yang dapat diterima. mengendalikan risiko yang paling penting langkah dalam pengelolaan risiko. Ini adalah faktor kunci untuk menentukan apakah manajemen risiko berhasil atau tidak. Tujuan dari pengendalian risiko keamanan e-government adalah untuk mengurangi tingkat risiko yang proyek e-government yang akan terjadi [ZZC08].

Ketiga jenis resiko diatas merupakan sebuah acuan untuk meminimalisasi resiko yang akan terjadi pada sistem *e-government.*

2.2.4. Pendekataan Centre of Excellence for Information Security

Pentingnya Teknologi Informasi (TI) pada skala individu, masyarakat, pemerintah dan perusahaan, dan bagaimana dampak ekonomi, kebijakan dan budaya. teknologi informasi dapat meningkatkan produk pendapatan domestik, meningkatkan produktivitas di banyak sektor. Cepatnya pertumbuhan dunia digital sekarang sangat rentan terhadap terus berubah yang mengakibatkan ancaman keamanan serius konsekuensi. Jika ancaman ini tidak ditangani cermat, implikasi dari layanan TI akan sangat terhambat. Keamanan informasi tidak terbatas pada perangkat lunak antivirus dan password, tetapi melampaui ini untuk sudut yang berbeda dan spektrum dari rekayasa sosial (*social engineering*) untuk

kriptanalisis (*cryptography*). Juga, keamanan informasi merupakan bagian integral dari keamanan nasional dan oleh karena itu investasi di bidang ini dari masyarakat pendidikan dan pelatihan berkualitas serta orang untuk membangun produk untuk misi kritis adalah sangat dituntut, sesuai dengan kebutuhan pada saat ini. Berikut pilar dari *Filosofis CoE for Information Security :*

Gambar II.27 Filosofi CoE for Information Security [KA08]

penjelasan :

gambar diatas menjelaskan, kolaborasi atau gabungan dari berbagai aspek. Mulai dai Pemerintah, Pendidikan, Lembaga Penelitian, Bisnis, yang dibagi dalam 3 sektor, yaitu : *Collaboration, Knowledge Transfer,* dan *Excellence* .Dimana ketiga entitas tersebut mempunyai *scope* dan area masing-masing [KAI08]

2.2.5. Proses Audit Keamanan Informasi Secara Sederhana (*Simple Information Security Audit Process)*

Keamanan di sistem apapun harus dinilai dalam hal risiko tersebut. Namun, proses untuk menentukan yang mengontrol keamanan sesuai dan biaya efektif

adalah cukup sering kompleks dan subjektif materi. Analisis Risiko merupakan komponen penting dalam mengamankan target perusahaan, dan dengan demikian memungkinkan risiko harus dikelola secara efektif. Cukup mengamankan perusahaan tidak bisa, namun dicapai tanpa menyeluruh Sistem Informasi audit keamanan, desain SISAP metodologi. The SISAP mengevaluasi kepatuhan perusahaan untuk keamanan dasar dan standar kontrol yang telah ditetapkan dalam kebijakan keamanan perusahaan dan hukum dan peraturan. SISAP ini, paling tidak mempunyai 3 objek atau fase dalam auditnya, antara lain :

- Menilai sejauh mana adalah perusahaan yang target audit (Tota) sesuai dengan ISO 17799
- Hitung perkiraan arus Tota postur keamanan dalam hal yang sesuai dengan praktek-praktek terbaik didefinisikan dalam ISO 17799
- Menghasilkan daftar klaim di berbagai bagian praktik keamanan terbaik didefinisikan dalam ISO 17799 yang akan melayani dalam mendefinisikan lingkup keamanan teknis tahap audit.

Tota adalah sebuah *metodology* audit keamanan informasi yang mengadopsi dari ISO 17799, sedangkan kerangka kerja minimal dari keamanan informasi adalah, mengadopsi dari TOTA tersebut. Mungkin saat ini sudah jarang mengadopsi dari standart tersebut tetapi, standar tersebut adalah dasar dari audit keamanan sistem informasi.

Berikut gambar dari Tota dan Nominal Keamanan :

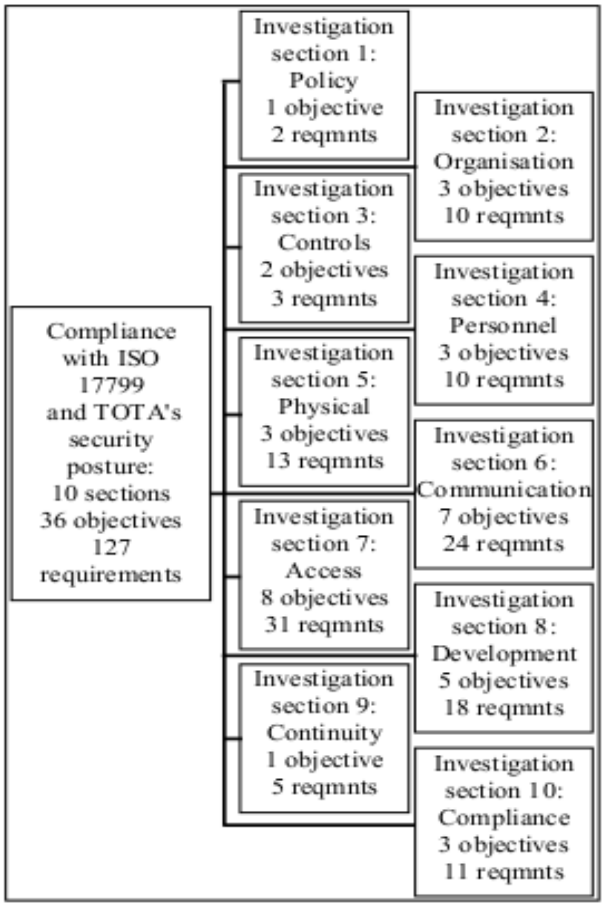


Figure 1: TOTA's compliance with ISO 17799

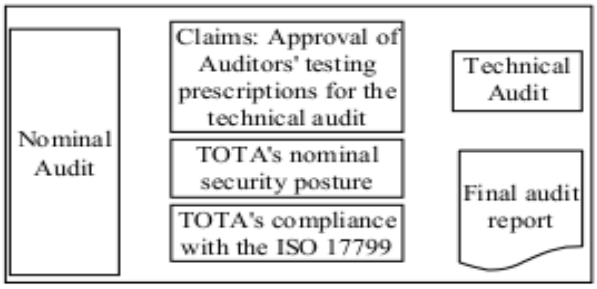


Figure 2: Framework for the nominal security

Gambar II. 28. Tota's and Framework Nominal Security[BE06]

Audit keamanan sistem informasi adalah satuhal yang wajib untuk mewujudkan keamanan sistem informasi dari suatu organisasi maupun perusahaan. Karena nilai keamanan suatu informasi adalah sebuat asset perusahaan yang tak ternilai [BEI06].

2.2.6. Kerangka Kerja Komponen Keamanan Informasi dan Antarmuka untuk Implementasi SSL (*Secure Socket Layer*)

Komponen keamanan informasi adalah independen komponen perangkat lunak dengan lebih dari satu fungsi ke mendukung layanan keamanan informasi pada tiap TI aplikasi daerah. Hal ini dirancang untuk memberikan inti keamanan informasi fungsi keamanan saat ini, seperti kerahasiaan, integritas, otentikasi, akses kontrol, dan *non-repudiation*. Beberapa komponen dari keamanan informasi, adalah sebagai berikut :

- **Kerahasiaan Layanan Komponen (*Confidentiality Component Service*)**, Kerahasiaan perlu untuk menyembunyikan penting informasi yang disimpan atau ditransmisikan dalam online dan offline lingkungan dari suatu yang tidak sah atau tidak teridentifikasi. Komponen ini menyediakan enkripsi dan dekripsi fungsi berdasarkan algoritma enkripsi konvensional dan algoritma enkripsi kunci publik.
- **Integritas Layanan Komponen (*Integrity Component Service*)**, Integritas diperlukan untuk melindungi konten informasi dikirimkan melalui jaringan dari yang ilegal diciptakan, diubah, atau dihapus. Komponen ini menyediakan hash algoritma dan MAC (*Message Authentication Code*) generasi fungsi.
- **Authentication Layanan Komponen (*Authentication Service Component*),** Otentikasi diperlukan untuk secara jelas mengidentifikasi entitas melalui pertukaran informasi. Komponen ini memberikan algoritma tanda tangan digital dan layanan integritas fungsi.
- **Kontrol Akses Layanan Komponen (*Control Access Service Component*),** Komponen ini menyediakan layanan otentikasi fungsi dan mencegah penggunaan yang tidak sah dari sumber daya dengan mengontrol akses berdasarkan hak akses.
- Non-repudiation Komponen Pelayanan (Non-Repudiation Service Component), Komponen ini digunakan untuk mencegah penolakan dari transmisi dan isinya antara pengirim dan penerima. Ini menyediakan fungsi layanan otentikasi.

Layanan komponen diatas adalah ada pada teknologi komunikasi yang terenkripsi untuk melindungi jalur komunikasi pada dunia internet yang saling berhubungan dan berkomuniikasi yaitu *Secure Socket Layer* (SSL) [JWS07]

2.3. Tinjuan Organisasi / Obyek Penelitian

Pada tinjauan organisasi atau obyek penelitian ini, Penulis mengambil obyek penelitian tersebut pada web portal Pemerintah Pusat yang terdiri dari kurang lebih berjumlah 56 web portal untuk Pemerintah Pusat saja. Kasus yang terjadi pada situs-situs web portal milik Pemerintah yang menjadi sasaran pada serangan dunia maya. Sepertihalnya, pada insiden *web defacement* pada tahun 2006 yaitu : **Protes Perang Libanon, Situs Pemda DKI disusupi** [DET06], dimana beritanya yaitu "Beberapa kelompok aktivis keamanan internet melakukan penyusupan pada situs resmi Pemerintah Provinsi DKI Jakarta. Bagian yang seharusnya menampilkan data statistik diubah menjadi halaman anti-perang. Foto seorang anak yang berdarah-darah tampil mencolok di halaman tersebut. Di bawah gambar itu, tertera kalimat berikut ini: *Lebanon-israel...STOP! No war I;ve Came Over Here To ask for peace... Nothing more nothing less. Life is already hard to live in with wars aing making easy The world was made to live in peace not to be in nuclear weapons STOP NOW.... PLZ..".* Takhanya itu, **Hacker Situs Golkar di Bekuk di Batam"** [DET06]**,** Polri akhirnya menangkap *hacker* situs Partai Golkar, Iqra Syafaat alias Nogra, pada 1 Agustus lalu" insiden itu terjadi pada tahun yang sama.

2.4. Kerentanan Sistem Informasi Pada Web Portal

Sistem adalah sifatnya tidak kekal akan selalu ada perubahan pada setiap waktunya yang mempengaruhinya. Kelemahan yang terdapat dalam Sistem Informasi memiliki dampak terhadap keamanan dari sistem itu sendiri yang disebut *Vulnerbiity*. *Vulnerbility* dapat terjadi akibat dari adanya *bug* atau kesalahan desain dalam sebuah sistem. Sebuah *vulnerbility* dapat menjadi hanya sekedar teori atau terbukti dengan adanya program eksploitaso dengan teknologi

fuzzing untuk melakukan eksploitasi terhadap *vulnerbility* tersebut.

Tulloch [TUL03] menjelaskam bahwa *vulnerbility* dapat terdiri dari apa saja yang mempengaruhinya, termasuk berikut ini :

- Kesalahan dalam melakukan konfigurasi pada sistem
- Cacat atau bug dari sistem operasi maupun aplikasi yang dijalankan oleh sebuah sistem
- Kelemahan yang menjadi bawaan dari protokol yang digunakan oleh layanan (*service*) pada target

Vulnerbility merupakan faktor utama yang sangat penting dalam merancang sebuah keamanan informasi. Dari indikator kelemahan tersebut kita bisa merancang sebuah keamanan yang cukup *powerfull* untuk membangunnya. Disini dalam mencari *vulnerbility* (kelemahan) pada suatu sistem berbeda caranya dan tergantung pada *scope* sistem yang akan kita bangun keamanannya tersebut.

2.4.1. OWASP (*Open Web Application Security Project*) Sepuluh Peringkat Resiko Kelemahan Web Aplikasi

OWASP merupakan sebuah organisasi yang bergerak dalam melakukan riset sebuah keamanan web aplikasi serta membuat sebuah *project* keamanan web aplikasi. OWASP adalah jenis organisasi baru, serta bebas dari tekanan komersial.. Tak hanya itu OWASP pun telah merilis Top 10 Resiko Keamanan Aplikasi pada tahun 2010, sebagai berikut :

A1- *Injection*

Kelemahan injeksi, seperti injeksi SQL, OS, dan LDAP, terjadi ketika data yang tidak dapat dipercaya dikirim ke suatu *interpreter* sebagai bagian dari suatu perintah atau *query*. Data berbahaya dari penyerang tersebut dapat mengelabui *interpreter* untuk mengeksekusi perintah yang tidak direncanakan, atau untuk mengakses data yang tidak terotorisasi.

A2 - *Cross Site Scripting* (XSS)

Kelemahan XSS terjadi ketika aplikasi mengambil data yang tidak dapat

dipercaya dan mengirimnya ke suatu *web browser* tanpa validasi yang memadai. XSS memungkinkan penyerang mengeksekusi *script-script* di dalam *browser* korban, yang dapat membajak sesi pengguna, mengubah tampilan website, atau mengarahkan pengguna ke situs-situs jahat.

A3 – *Broken Authentication and Session Management*

Fungsi-fungsi aplikasi yang berhubungan dengan otentikasi dan pengelolaan sesi seringkali tidak dimplementasikan dengan benar. Hal ini memungkinkan penyerang mendapatkan *password, key,* Pengelolaan Sesi dan token-token sesi, atau mengeksploitasi cacat implementasi lainnya untuk memperoleh identitas yang buruk pengguna yang lain .

A4 - *Insecure Direct Object Management*

Direct object reference terjadi ketika pengembang mengekspos referensi ke suatu objek implementasi internal, seperti file, direktori, atau kunci database. Tanpa adanya suatu pemeriksaan kendali akses atau perlindungan lainnya, penyerang dapat memanipulasi referensi-referensi ini untuk mengakses data yang tidak terotorisasi.

A5 - *Cross Site Request Foregery* (CSRF)

Suatu serangan CSRF memaksa *browser* korban yang sudah *log-on* untuk mengirim HTTP request yang dipalsukan, termasuk di dalamnya session *cookie* korban dan informasi otentikasi lain yang otomatis disertakan, ke suatu aplikasi web yang rentan. Hal ini memungkinkan penyerang untuk memaksa browser korban menghasilkan request yang dianggap sah oleh aplikasi rentan tadi.

A6 - *Security Miss Configuration*

Keamanan yang baik mensyaratkan dimilikinya suatu konfigurasi keamanan (yang terdefinisi dan diterapkan) untuk aplikasi, framework, server aplikasi, *web server*, *server database*, dan *platform*. Semua pengaturan ini harus didefinisikan, diimplementasikan,dan dipelihara, karena terdapat banyak aplikasi yang dirilis tanpa konfigurasi default yang aman. Hal ini juga mencakup menjaga semua software *up-to-date*, termasuk semua pustaka kode yang digunakan aplikasi tersebut.

A7 - *Failure Restric URL Access*

Banyak aplikasi web yang tidak melindungi data sensitif (seperti data kartu kredit, SSN, kredensial otentikasi) dengan enkripsi atau *hashing* yang memadai. Penyerang dapat mencuri atau memodifikasi data dengan perlindungan lemah semacam itu untuk melakukan pencurian identitas, kejahatan kartu kredit, atau kriminalitas lain.

A8 - *Unvalidated Redirects and Forward* (*New*)

Banyak aplikasi web memeriksa hak akses URL sebelum memberikan link dan tombol-tombol yang diproteksi. Bagaimanapun juga, aplikasi perlu melakukan pemeriksaan kendali akses yang serupa setiap kali halaman-halaman ini diakses, atau penyerang akan dapat memalsukan URL untuk mengakses halaman-halaman yang tersembunyi ini

A9 -*Insecure Cryptographic Storages*

Aplikasi seringkali gagal untuk mengotentikasi, mengenkripsi, dan melindungi kerahasiaan serta integritas lalu-lintas jaringan yang sensitif. Ketika aplikasi gagal melakukan hal-hal tersebut, adalah dikarenakan ia mendukung algoritma yang lemah, menggunakan sertifikat yang tidak *valid* atau sudah kadaluarsa, atau karena tidak menggunakannya dengan benar .

A10 -*Insufficient Transport Layer Protection*

Aplikasi web seringkali mengarahkan (*redirect*) dan meneruskan (*forward*) pengguna ke halaman data yang tidak menentukan halaman *Forward* yang Tidak dan website lain, dan mengunakan penyerang dapat dapat dipercaya untuk ke situs *phishing* atau tujuan. Tanpa validasi yang tepat, mengarahkan korban divalidasi *malware*, atau menggunakan *forward* untuk mengakses halaman yang tidak terotorisasi.

Semua resiko keamanan diatas merupakan sebagian besar dari kelemahan yang terdapat pada *web server*, dimana *web server* adalah sebuah layanan http yang berfungsi untuk meneruskan permintaan dari suatu client untuk menampilkan halaman web dari sebuah *web server*. Jenis *web server* disini bermacam-macam tergantung kebutuhan sebuah organisasi tersebut, tetapi sebagian besar penduduk

di dunia ini menggunakan *web server* apache. Kita juga mengetahui sebenarnya produk suatu *web server* adalah banyak macamnya seperti : Apache *web server*, Tomcat, Nginx, IIS *web server*, dan lain sebagainya.

2.5. Pola Fikir

Penetration Testing merupakan sebuah metode untuk melakukan evaluasi keamanan dari Sistem Komputer atau jaringan dengan melakukan simulasi serangan yang dilakukan oleh *Hacker* [WP05]. Proses dari *Penetration Testing* adalah berusaha untuk mengumpulkan dan menganalisa kelemahan dari sistem, kesalahan teknis dan kerentanan yang mungkin ada. Tujuan dari Penetration Testing ini adalah melakukan identifikasi dari ekplorasi dan kelemahan yang terdapat dalam organisasi dan untuk membantu memastikan efektifitas (atau tidak) dari tingkat keamanan yang telah diimplemtasikan [MID05].

Hal yang harus perlu diperhatikan dalam proses penetration testing adalah bahwa model yang digunakan dalam melakukan Penetration Testing haruslah sedapat mungkin menyerupai serangan yang sesungguhnya. Midian [MID05] menjelaskan bahwa ada dua macam model yang dapat dipergunakan dalam melakukan *Penetration Testing,* yaitu *Zero Knowledge Test* (*Black Box*) dan *Full Knowledge Test* (*White Box*). Pada model pertama seorang penetration tester tidak tahu menahu semua informasi sistem pada target sehingga tidak fokus pada satu target saja, akan tetapi dalam sistem keamanan justru di lain target mempengaruhi kelemahan sistem tersebut. Sedangkan model ke dua, seseorang yang akan melakukan *Penetration Testing* akan diberikan informasi yang lengkap tentang target tersebut. Kedua model ini digunakan oleh organisasi setelah terlebih dahulu melakukan identifikasi ancaman yang terjadi terhadap sistem informasi. Sehingga Penulis mempuyai pola fikir, sebagai berikut :

2.5.1. Sisi Hukum dari *Penetration Testing*

Penetration Testing harus dilakukan oleh orang yang berkompeten dalam bidang tersebut yang biasanya adalah konsultan. Dalam melakuakn proses *Penetration Testing*, seorang konsultan harus terikat dengan perjanjian antara

organisasi yang akan diuji dan perusahaan konsultan tersebut. Klevisky, Laliberte & Gupta [KLE02] menjelaskan bahwa keseluruhan proses dari *Penetration Testing* dibatasi oleh peraturan – peraturan yang telah ditetapkan dalam perjanjian. Dari sisi sinilah proses *Penetration Testing* dianggap sebagai sebuah kegiatan legal, meskipun menggunakan cara-cara ilegal yang biasa digunakan oleh *cracker*. Terkadang yang masih masalah adalah tidak ada satu alat pun yang mampu menjamin tingkat keamanan client dalam proses *Penetration Testing.* Hal ini sama dengan sebuah semboyan "*Security Is Process*" dalam hal ini keamanan adalah sebuah proses untuk mengantisipasinya, karena dalam ilmu keamanan informasi belum ada yang menjudge bahwa itu aman. Dan *Security* juga bukan sebuah teknologi yang harus kita beli secara mahal. *Security* adalah diri kita sendiri dengan cara pandang kita masing-masing dengan cara ingin belajar, *aware* dan *care* terhadap *issue* keamanan informasi yang ada

.

2.6. Hipotesis

Dalam rangka melakukan penelitian terhadap objek keamanan informasi dari instansi Pemerintah Pusat terkait, maupun terhadap alat ukur (*penetration testing*), Penulis merumuskan hipotesis untuk dijadikan asumsi awal dalam penelitian ini. Nasution [NAS04] mengemukakan bahwa hipotesis adalah pernyataan tentatif yang merupakan dugaan atau terkaan tentang apa saja yang kita amati dalam usaha dalam memahaminya. Namun disini harus dibedakan antara hipotesis dan teori, meskipun pada permulaannya teori bisa berupa hipotesa dari seseorang namun toeri tetap bertujuan untuk mengatur fakta-fakta dan memberinya makna. Oleh karena itu, secara singkat hipotesis memiliki fungsi [NAS04] :

- Menguji kebenaran suatu teori
- Memberi ide untuk mengembangkan teori
- Memperluas pengetahuan tentang gejala-gejala yang kita pelajari

2.6.1. Hipotesis Penelitian

Apabila model untuk keamanan web portal Pemerintah ini dibangun, maka keamanan informasi pada web portal Pemerintah akan terminimalisasi terhadap resiko-resiko ancaman yang ada pada web portal. Dengan menggunakan model *web application firewall* diperkirakan akan sesuai digunakan untuk model keamanan web portal Pemerintah.