BAB 2

TINJAUAN PUSTAKA

2.1. Tinjauan Perusahaan

2.1.1. Latar Belakang, Misi dan Sertifikasi Perusahaan

XecureIT (PT. IMAN Teknologi Imformasi) adalah mitra terpercaya dan handal dalam keamanan informasi sejak tahun 2005. XecureIT diciptakan oleh *Information Security Profesional* yang memiliki kemampuan, pengetahuan, dan pengalaman lebih dari 20 tahun di seluruh dunia.

Misi XecureIT adalah untuk melakukan layanan PERAWATAN keamanan informasi dengan menggunakan keahlian para professional yang terpercaya, baik bagi masyarakat, maupun pelanggannya.

Kombinasi semangat antara XecureIT *standard security framework* (XGEF) dan keahlian tim memastikan bahwa penyampaian informasi secara efektif dapat membantu *client* untuk meningkatkan keamanan informasi yang dianggap berharga, serta dengan adanya penggunaan *critical IT systems* yang berfungsi untuk menjaga efisiensi biaya yang akan dikeluarkan.

Keinginan kuat yang dimiliki oleh XecureIT dalam melindungi rahasia pelanggannya telah dibuktikan dengan menjadi perusahaan penyedia jasa keamanan informasi pertama di Indonesia yang menerapkan keamanan tingkat tinggi, dimana semua layanan telah disertifikasi dengan ISO 27001. Serta manajemen keamanan sistem informasi yang telah terakreditasi secara langsung oleh bythe British Standards Institution (BSI).

2.1.2. Security Services: CARE

XecureIT memberikan layanan keamanan informasi yang lengkap, yang dikenal sebagai CARE ***(C**onsultancy – **A**ssurance - **R**esearch - **E**ducation)*

1. *Information Security Consultancy*
   A. Mengembangkan strategi keamanan informasi, seperti kebijakan, prosedur dan arsitektur.
   B. Membantu agar keamanan informasi sesuai dengan standar internasional, seperti:
      a. ISO 20000 Information Technology Service Management (ITSM)
      b. ISO 22301 Business Continuity Management System (BCM)
      c. ISO 27001 Information Security Management System (ISMS)
      d. PCI DSS (Payment Card Industry Data Security Security Standard)
      e. TIA-942 Data Center Standards
   C. Mengimplementasikan *system security hardening* dan *awareness program*
   D. Melakukan penanganan insiden dan menginvestigasi *cyber crime*
2. *Information Security Assurance*
   A. Melakukan audit*, assessment* and *penetration testing*
   B. Menyediakan layanan pengelolaan keamanan, seperti administrasi sistem keamanan IT,
3. *Information Security Research*

   White Hacker Lab, R&D Division, yang meneliti masalah terkait keamanan informasi, menganalisa trend keamanan saat ini, menemukan teknik *hacking* yang baru dan mengembangkan solusi ofensif dan defensif perusahaan untuk *client* dalam menghadapi tantangan sistem keamanan informasi.

   Dalam penelitian yang telah dilakukan oleh XecureIT tentang internet banking, membuktikan bahwa sangat mudah untuk melakukan perampokan pada iBank, meskipun transaksi yang terjadi dalam lingkup iBank dilindungi dengan menggunakan *hardware security token.* Oleh karena itu, XecureIT mengembangkan *XecureBrowser* (XB) sebagai solusi *open source* gratis bagi nasabah bank.

4. *Information Security Education*

Berikut adalah layanan yang disediakan oleh *SecurityFirst, Division of Education* dan *Community Development*:

A. Pelatihan profesional berkualitas tinggi dan sertifikasi, seperti:
   a. Certified Information Systems Security Professional (CISSP)
   b. Certified Information Systems Auditor (CISA)
   c. Certified ISO 27001 Lead Auditor
   d. ISO 27001 Lead Implementer
   e. Advance Penetration Testing (General, Web Application & WiFi)
   f. Advance Hacking Countermeasures (General, Web Application & WiFi)
B. Pengembangan tim keamanan internal melalui pelatihan dalam rumah.
C. kampanye kesadaran publik melalui pelatihan berbasis web secara gratis, poster, artikel dan seminar.

2.1.3. Company Value: PASSION

Keinginan XecureIT untuk menyediakan layanan CARE yang berharga bagi *client*. Keinginan (tujuan) XecureIT adalah:

1. ***P****rofessional value is TRUSTED*
2. ***A****ccountable professional is guaranteed*
3. ***S****ecure professional services*
4. ***S****haring our professional knowledge and experience*
5. ***I****nformative report and explanation*
6. ***O****n earth recommendations*
7. ***N****etworking with other information security professionals*

2.1.4. Professional: Experience, TRUSTED Value and Certification

Tim XecureIT terdiri dari para profesional dengan keterampilan, pengetahuan, dan pengalaman lebih dari 20 tahun di seluruh dunia dalam hal IT dan keamanan

informasi. Tim XecureIT menyelesaikan dan menangani berbagai macam tantangan keamanan informasi di negara-negara seperti Eropa, Amerika, Asia dan Afrika yang memiliki berbagai macam tempat dan kondisi sehari-hari yang berbeda. TRUSTED (***T**ruth - **R**eliable - **U**seful - **S**ecret - **T**imely - **E**xpert – **D**edicated*) adalah nilai profesional yang menjadi fondasi yang kuat dari layanan CARE.

Profesional-profesional XecureIT memiliki *vendor neutral international certifications*, seperti:

1. Certified Information Systems Auditor (CISA)
2. Certified Information Systems Security Professional (CISSP)
3. IRCA Certified ISO 27001 Lead Auditor
4. Certified Incident Handling (CIH)
5. Computer Hacking Forensic Investigator (CHFI)

2.1.5. Logo Perusahaan



Gambar 2.1 Logo Perusahaan XecureIT

2.2. Landasan Teori

2.2.1 Internet

Internet, kependekan dari *interconnection-networking*, secara harfiah ialah sistem global dari seluruh jaringan komputer yang saling terhubung menggunakan standar Internet Protocol Suite (TCP/IP) untuk melayani miliaran pengguna di seluruh dunia. TCP/IP digunakan sebagai protokol pertukaran paket (*packet switching communication protocol*). Rangkaian jaringan komputer yang terbesar dinamakan Internet.

Internet merupakan jaringan komputer yang dibentuk oleh Departemen Pertahanan Amerika Serikat di tahun 1969, melalui proyek ARPA yang disebut ARPANET (*Advanced Research Project Agency Network*), dimana ARPA mendemonstrasikan bagaimana dengan hardware dan software komputer yang berbasis UNIX dapat melakukan komunikasi dalam jarak yang tidak terhingga melalui saluran telepon.

Proyek ARPANET merancang bentuk jaringan, kehandalan, seberapa besar informasi dapat dipindahkan, dan akhirnya semua standar yang ARPA tentukan menjadi cikal bakal pembangunan protokol baru yang sekarang dikenal sebagai TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Tujuan awal dibangunnya proyek tersebut adalah untuk keperluan militer. Pada saat itu Departemen Pertahanan Amerika Serikat (*US Department of Defense*) membuat sistem jaringan komputer yang tersebar dengan menghubungkan komputer di daerah-daerah vital untuk mengatasi masalah bila terjadi serangan nuklir dan untuk menghindari terjadinya informasi terpusat, yang
apabila terjadi perang dapat mudah dihancurkan.

Pada mulanya ARPANET hanya menghubungkan 4 situs saja yaitu Stanford Research Institute, University of California, Santa Barbara, University of Utah, dimana ARPANET membentuk satu jaringan terpadu di tahun 1969, dan secara umum ARPANET diperkenalkan pada bulan Oktober 1972. Tidak lama kemudian

proyek ini berkembang pesat di seluruh daerah, dan semua universitas di negara tersebut ingin bergabung, sehingga membuat ARPANET kesulitan untuk mengaturnya.

Oleh sebab itu ARPANET dipecah manjadi dua, yaitu "MILNET" untuk keperluan militer dan "ARPANET" baru yang lebih kecil untuk keperluan nonmiliter seperti, universitas-universitas. Gabungan kedua jaringan akhirnya dikenal dengan nama DARPA Internet, yang kemudian disederhanakan menjadi Internet.

2.2.2. Situs

Sebuah situs web sering pula disingkat situs; *website*, *site* adalah sebutan bagi sekolompok halaman (*domain name*) atau subdomain di *World Wide Web* (WWW) di internet. WWW terdiri dari situs web diakses dari sebuah URL yang menjadi "akar" (*root*), yang disebut *homepage* (halaman induk sering diterjemahkan menjadi "beranda", "halaman muka"), dan biasanya disimpan dalam server yang sama. Tidak semua situs web dapat diakses dengan gratis beberapa situs web melakukan pembayaran agar dapat menjadi pelanggan misalnya situs-situs berita, layanan surat elektronik (e-mail), dan lain-lain.

Terminologi dari website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain didalam World Wide Web (WWW) pada internet. Sebuah *web page* adalah sebuah dokumen yang tertulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website-website tersebut dapat membentuk sebuah jaringan informasi yang sangat besar. Halaman-halaman dari website akan bisa diakses melalui URL yang biasa disebut homepage. URL ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun, *hyperlink* yang ada dihalaman tersebut mengatur para pembaca dan memberitahukan pembaca susunan keseluruhan dan bagian arus informasi ini berjalan. Beberapa website membutuhkan subskripsi (data masukan) agar para *user* bisa mengakses situs tersebut.

Penemu website adalah Sir Timothy John “Tim” Berners-Lee, sedangkan website yang tersambung dengan jaringan, pertamakali muncul pada tahun1991. Maksud dari Tim ketika membuat website adalah untuk mempermudah tukar19 menukar dan memperbaharui informasi kepada sesama peneliti di tempat dia bekerja. Pada tanggal 30 April 1993, CERN (tempat dimana Tim bekerja) menginformasikan bahwa WWW dapat digunakan gratis oleh semua orang. Sebuah website bisa berupa hasil kerja dari perorangan atau individu, atau menunjukkan kepemilikan dari sebuah organisasi, perusahaan, dan biasanya website itu menunjukkan beberapa topik khusus, atau kepentingan tertentu. Sebuah website bisa berisi hyperlink yang menghubungkan ke website lain.

Website ditulis, atau secara dinamik dikonversikan menjadi HTML dan diakses melalui sebuah program software yang bisa disebut dengan web browser, yang dikenal juga dengan HTTP Client. Halaman web dapat dilihat atau diakses melalui jaringan komputer dan internet, perangkatnya bisa saja berupa personal komputer, laptop komputer, *PDA* ataupun *Cellphone*. Sebuah website dibuat didalam sebuah sistem komputer yang dikenal dengan web server juga disebut dengan HTTP Server, dan pengertian ini juga bisa menunjukkan pada software yang dipakai.

Sebuah website dinamis adalah website yang secara berkala, informasi didalamnya berubah, atau website ini bisa berhubungan dengan *user* dengan berbagai macam cara atau metode (HTTP cookies atau Variabel Database, sejarah kunjungan, variabel sesi dan lain-lain) bisa juga dengan cara interaksi langsung menggunakan *form* dan pergerakan mouse. Ketika web server menerima permintaan dari *user* untuk memberikan halaman tertentu, maka halaman tersebut akan secara otomatis diambil dari media penyimpanan sebagai respon dari permintaan yang diminta oleh user. Sebuah situs dapat menampilkan dialog yang sedang berlangsung diantara dua user, memantau perubahan situasi, atau menyediakan informasi yang berkaitan dengan sang user. Ada banyak jenis sistem software yang dapat dipakai untuk meng-*generate Dynamic Web System* dan Situs Dinamis, beberapa diantaranya adalah ColdFusion (CFM), Active Server Pages (ASP), Java Server Pages (JSP) dan PHP, bahasa program yang mampu untuk meng-*generate Dynamic Web System* dan Situs Dinamis.

Situs juga bisa termasuk didalamnya berisi informasi yang diambil dari satu atau lebih database atau bisa juga menggunakan teknologi berbasis XML, contohnya adalah RSS. Isi situs yang statis juga secara periodik di-*generate*, atau, apabila ada keadaan dimana dia butuh untuk dikembalikan kepada keadaan semula, maka dia akan di-*generate*, hal ini untuk menghindari kinerjanya supaya tetap terjaga.

Seperti yang tertulis diatas, terdapat perbedaan dalam penulisan dari terminologi website. Walaupun ¨Website¨ sudah secara umum dipakai, namun untuk Associated Press Stylebook, Reuters, Microsoft, Academia, dan kamus-kamus yang ada, penulisan yang dipakai adalah dengan menggunakan dua kata, yaitu Web Site. Hal ini karena Web bukanlah terminologi umum, namun ia adalah kependekan dari World Wide Web.

2.2.3. HTTP Respon

Request dari klien ditangani oleh server dan diresponnya secara tepat. Dalam direspon, server mengirimkan kembali serangkaian komponen pesan yang dapat dikategorikan menjadi: [1]

1. Kode Respon

   Kode Respon adalah sebuah kode angka yang sesuai dengan respon yang bersangkutan. Kode Respon merupakan bagian awal dari respon server dan menetapkan nada untuk mengingatkan interaksi tersebut. Server merespon dalam salah satu dari empat cara ini: *Success, Redirection, Client Error,* atau *Server Error*. Setiap respon memiliki arti yang berbeda satu sama lain, tergantung permintaan klien. Tabel 2.1 menunjukkan kode-kode respon umum yang sering digunakan.

Tabel 2.1 *Kode-kode Respon Umum dan Respon-responnya*

Kode Respon	**Keterangan**
Success 2xx	
200 OK	*Request* berhasil dijalankan

Redirection 3xx	
301 Moved Permanently	Sumber-sumber *request* diberi URL permanen dan ditempatkan pada field location. Kode respon ini berkata, "saya telah dipindahkan, ikuti saya ke tempat yang baru."
302 Moved Permanently	Sumber-sumber *request* telah diberi URL temporer yang baru dan ditempatkan pada field location. Kode respon ini berkata, "saya telah dipindahkan, ikuti saya ke tempat sementaraku yang baru, tetapi jangan bergantung pada saya jika ingin disini dalam waktu lama."
Client Error 4xx	
400 Bad Request	*Request* tidak dimengerti oleh server.
401 Unauthorized	Sumber-sumber yang diminta memerlukan pembuktian keaslian user, biasanya dalam bentuk pembuktian keaslian Basic atau yang sejenis.
403 Forbidden	Server memahami *request* tetapi menolak untuk merespon. Khususnya sewaktu metode GET digunakan untuk menerima respon ini, sedikit atau tak ada informasi lebih jauh lagi. Tetapi, sewaktu metode HEAD dipakai, beberapa server akan member informasi rinci lebih lengkap mengenai mengapa kondisi seperti ini muncul.
404 Not Found	Sumber-sumber yang diminta tak ditemukan.
Server Errors 5xx	
500 Internal Server Error	Server menemukan kesalahan internal

	(internal error) dalam memproses *request*.
501 Not Implemented	Server tidak mendukung *request* yang bersangkutan.
502 Bad Gateway	Server menerima respon cacat dari server utama sewaktu ia mengirim resource yang diminta. Respon ini khusus untuk proxy-proxy HTTP.
503 Service Unavailable	Server tidak mampu merespon *request* karena *request* sedang meluap.

Kode respon 501 (Not Implemented) adalah cara server memberitahu kepada klien bahwa metode yang diminta tidak didukung. Ini terjadi, misalnya jika mengirim *request* yang diminta dengan metode OPTIONS kepada server HTTP/1.0, padahal OPTIONS hanya didukung oleh HTTP/1.1

2. Field Header

Field Header merupakan informasi tambahan mengenai respon tersebut. Respon server atas *request* klien dan respon klien atas *request* server, berisi field-field header yang mengandung beragam tingkatan informasi tambahan. Server dan klien membagi-bagi field-field ini dan menggunakan informasi sesuai kebutuhan. Tabel 2.2 berisi definisi field header utama.

Tabel 2.2 *Definisi Field Header*

Field Header	**Keterangan**
Allow	Menampilkan metode-metode yang didukung oleh sumber-sumber yang diminta.
Authorization	Menampilkan daftar mandat-mandat otorisasi untuk pembuktian keaslian HTTP.
Content-Encoding	Menampilkan *encoding* isi tambahan yang

	sedang dijalankan pada pengembalian data. Dengan informasi ini klien mengetahui secara lebih baik bagaimana menginterpretasikan data yang dikembalikan. Misalnya, Content-Encoding: x-gzip berarti isi gzip terkompresi.
Content-Length	Menampilkan daftar ukuran body isi dalam angka decimal basis delapan. Misalnya: Content-Length: 332.
Content-Type	Menampilkan tipe isi dalam respon. Misalnya Content-Type:text/html menampilkan text/html sebagai tipe isi. Field ini membantu klien dalam memahami secara lebih baik bagaimana menampilkan isi dalam browser. Date Menampilkan daftar tanggal dan jam server.
Expires	Menampilkan daftar tanggal dan jam yang menandakan isi telah out of date.
From	Menampilkan daftar alamat e-mail yang dipakai untuk mengidentifikasi bagian respon isi. Field ini jarang digunakan.
Last-Modified	Menampilkan daftar tanggal dan jam yang diyakini server sebagai saat terakhir sumber-sumber yang diminta dimodifikasi.
Location	Menampilkan daftar lokasi sumber-sumber yang diminta.
Pragma	Menggambarkan perilaku tambahan untuk *request*. Misalnya, bila field header Pragma dikirim dari server dengan perintah “no-cache”, klien harus me-load isi yang dikirim apakah iasudah meng-cache salinannya. Contoh: Date:

	Thu, 22 Apr 2002 01:10:22 GMT.
Referrer	Mengizinkan klien untuk menentukan alamat sumber. Server Daftar software yang berjalan pada server. Dalam banyak hal, informasi ini akurat. Misalnya, Server: Microsoft-IIS/5.0. Tetapi, administrator yang pintar bisa mengubah informasi ini sesukanya, misalnya "Web Server Saya."
User-Agent	Daftar informasi tambahan mengenai user agent (klien) yang meminta informasi. Misalnya, User-Agent:Mozilla/5.0 (WinNT).
WWW-Authenticate	Digunakan untuk merespon kode respon 401 *Unauthorized*. Field ini menjaga tak masuknya akses ilegal.

3. Data

   Data merupakan isi atau badan (body) dari respon. Bagian data pada *request* klien atau pada respon server sungguh merupakan badan dari komunikasi antara keduanya. Misalnya bila menggunakan metode *request* GET untuk sumber-sumber default, perintah berikut ini yang dijalankan:

   C:\ nc.exe www.informasimu.com 80 GET

   /HTTP/1.0

   Another <cr><lf>here

   dan halaman web default (data atau body) akan dikirim kembali.

Dengan tiga komponen tersebut, browser klien memahami respon server dan berinteraksi dengan server.

2.2.4 Keamanan Komputer

Sistem keamanan komputer digunakan untuk menjamin agar sumber daya tidak digunakan atau dimodifikasi orang yang tidak diotorisasi. Pengamanan termasuk masalah teknis, manajerial, legalitas dan politis. Kemanan sistem terbagi menjadi tiga, yaitu: [2]

1. Keamanan eksternal adalah pengamanan yang berhubungan dengan fasilitas komputer dari penyusup dan bencana, misalnya bencana alam.
2. Keamanan interface pemakai, berkaitan dengan identifikasi pemakai sebelum diizinkan mengakses program dan data yang tersimpan didalam sistem.
3. Keamanan internal, berkaitan dengan beragam pengamanan yang dibangun pada perangkat keras dan sistem operasi untuk menjamin operasi yang handal dan untuk menjaga keutuhan program serta data. Sehubungan dengan keamanan ini terdapat dua masalah yang penting, yaitu:
   A. Kehilangan data, yang dapat disebabkan antara lain oleh:
      a. Bencana, seperti kebakaran, banjir, gempa bumi, perang, maupun kerusuhan.
      b. Kesalahan perangkat keras dan perangkat lunak yang disebabkan oleh tidak berfungsinya pemroses, disk yang tidak terbaca, kesalahan telekomunikasi, dan kesalahan program (*bugs*).
      c. Kesalahan manusia, seperti salah dalam memasukkan data, salah memasang disk, eksekusi program yang salah, dan kehilangan disk.
   B. Penyusup (*intruder*) yang terdiri dari:
      a. Penyusup pasif, yaitu membaca data yang tidak diotorisasi.
      b. Penyusup aktif, yaitu mengubah data yang tidak diotorisasi. Ketika hendak merancang sebuah sistem yang aman dari serangan para *intruder*, adalah penting untuk mengetahui sistem tersebut akan dilindungi dari intruder apa. Empat contoh kategori:
         1) Keingintahuan seorang akan hal-hal pribadi orang lain.

Banyak orang yang mempunyai PC yang terhubung ke jaringan. Beberapa orang dalam jaringan tersebut suka membaca e-mail dan file orang lain jika didalam jaringan tersebut tidak ditempatkan system penghalang. Sebagai contoh, sebagian besar sistem UNIX mempunyai default bahwa semua file yang baru diciptakan dapat dibaca oleh orang lain.

2) Penyusupan oleh orang-orang dalam.

   Pelajar, programmer, operator, dan personil teknis menganggap bahwa mematahkan sistem keamanan komputer lokal merupakan suatu tantangan.

3) Keinginan untuk mendapatkan uang.

   Beberapa pemrogram bank mencoba mencuri uang dari bank tempat pemrogram tersebut bekerja dengan mengubah software sehingga akan memotong bunga daripada membulatkannya, menarik uang dari akun yang sudah tidak digunakan selama bertahun-tahun, ataupun memeras ("Bayar saya, atau saya akan menghancurkan semua record bank Anda").

4) E-spionase komersial atau militer.

   E-spionase adalah usaha serius yang diberi dana besar oleh pesaing atau pihak musuh untuk mencuri program, rahasia dagang, ide-ide paten, teknologi, rencana bisnis, dan sebagainya. Seringkali usaha ini melibatkan *wiretapping*, dimana antena diarahkan ke suatu computer untuk menangkap radiasi elektromagnetis yang memancar dari komputer itu. Perlindungan terhadap rahasia militer suatu negara dari pencurian oleh negara lain sangat berbeda dengan perlindungan terhadap pelajar yang mencoba memasukkan *message-of-the-day* pada suatu sistem. Terlihat jelas bahwa jumlah usaha yang berhubungan dengan keamanan dan proteksi tergantung pada siapa musuhnya. Secara garis besar pengamanan sistem komputer mencakup empat hal yang sangat mendasar, yaitu:

a) Pengamanan Fisik

Pengamanan fisik dapat dilakukan dengan menempatkan sistem komputer pada tempat yang mudah diawasi dan dikontrol untuk mengantisipasi kelalaian/keteledoran dari para user yang sering meninggalkan terminal komputer dalam keadaan *logon*. Keteledoran semacam ini memungkinkan pihak lain untuk dapat mengakses beberapa fasilitas sistem komputer yang sebenarnya bukan menjadi hak para pengakses tersebut. Bahkan para pengakses yang tidak memiliki hak tersebut dapat melakukan tindakan perusakan terhadap sistem komputer tersebut.

b) Pengamanan Akses

Pengamanan akses biasanya menjadi tanggung jawab para administrator sistem. Dalam hal ini seorang administrator harus mampu mengontrol dan mendokumentasikan seluruh akses ke sistem komputer dengan baik sehingga dapat mempercepat dan mempermudah pencarian penyebab masalah dan solusinya bila suatu saat terjadi sesuatu dalam system komputer tersebut.

c) Pengamanan Data

Pengamanan data dilakukan dengan menerapkan sistem tingkatan akses dimana seseorang hanya dapat mengakses data tertentu saja yang menjadi haknya. Sebagai contoh, departemen pemasaran dari suatu perusahaan hanya dapat mengakses data yang berkaitan dengan pemasaran barang dan tidak dapat mengakses data gaji pegawai karena data gaji pegawai merupakan wewenang departemen personalia. Data yang sangat sensitif dapat dipasangi password tertentu.

d) Pengamanan Jaringan

Jaringan disini berkaitan erat dengan pemanfaatan jaringan publik/umum seperti Internet. Pengamanan jaringan dapat dilakukan dengan menggunakan kriptografi dimana data yang sifatnya sensitive dienskripsi terlebih dahulu sebelum ditransmisikan melalui internet. Dengan mentransmisikan data yang telah dienskripsi maka walaupun data tersebut jatuh ke pihak yang tidak berhak, pihak tersebut tidak akan dapat mengerti isinya. Setelah mengetahui keempat cakupan dalam keamanan system komputer maka diharapkan akan semakin menyadari pentingnya pengamanan sistem komputer sehingga menjadi lebih serius lagi didalam menangani danmengelolanya.

2.2.4.1 Ancaman Keamanan

Target pengamanan adalah menghindari, mencegah, dan mengatasi ancaman-ancaman terhadap sistem. Kebutuhan akan pengamanan komputer dapat dikategorikan dalam tiga aspek, yaitu:

1. Kerahasiaan, dimana informasi pada sistem komputer itu terjamin kerahasiaannya, hanya dapat diakases oleh pihak-pihak yang diotorisasi, keutuhan serta konsistensi data pada sistem tersebut tetap terjaga.
2. Integritas, dimana sumber daya sistem terjamin hanya dapat dimodifikasi oleh pihak-pihak yang diotorisasi.
3. Ketersediaan, adalah sumber daya sistem komputer terjamin akan tersedia bagi pihak-pihak yang diotorisasi pada saat diperlukan.

2.2.4.2 Tipe-tipe Ancaman Keamanan

Fungsi sistem komputer dapat digunakan sebagai dasar untuk menentukan model tipe ancaman dari suatu sistem komputer. Berdasarkan fungsi system komputer sebagai penyedia informasi, ancaman terhadap sistem computer dikategorikan menjadi empat, yaitu:

1. *Interruption,* merupakan suatu ancaman terhadap *availability,* informasi atau data yang ada dalam sistem komputer dirusak, dihapus, sehingga jika dibutuhkan maka sudah tidak ada lagi.
2. *Interception,* merupakan ancaman terhadap kerahasiaan (*secrecy*). Informasi yang ada didalam sistem disadap oleh orang yang tidak berhak.
3. *Modification,*merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil menyadap lalu-lintas informasi yang sedang dikirim lalu mengubahnya sesuai keinginan orang itu.
4. *Fabrication,* merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil meniru atau memalsukan suatu informasi sehingga orang yang menerima informasi tersebut menyangka informasi tersebut berasal dari orang yang dikehendaki oleh si penerima informasi tersebut.

2.2.4.3 Tipe Ancaman Pada Aplikasi Web

Berikut ini adalah berbagai jenis model serangan yang terjadi pada aplikasi web:

1. SQL Injection

   Pada dasarnya SQL Injection merupakan cara mengeksploitasi celah keamanan yang muncul pada level atau "layer" database dan aplikasinya. Celah keamanan tersebut ditunjukkan pada saat penyerang memasukkan nilai "string" dan karakter-karakter contoh lainnya yang ada dalam instruksi SQL; perintah tersebut hanya diketahui oleh sejumlah kecil individu yang berusaha untuk mengeksploitasinya. Karena tipe data yang dimasukkan tidak sama dengan yang seharusnya, maka terjadi sebuah aktivitas "liar" yang tidak terduga yang dapat mengakibatkan individu yang tidak berhak dapat masuk ke dalam system yang telah terproteksi. Dikatakan sebagai sebuah "injeksi" karena aktivitas penyerangan dilakukan dengan cara memasukkan *string* khusus untuk melewati filter logika hak akses pada website atau system komputer yang dimaksud.

2. Cross Site Scripting

   Cross Site Scripting (XSS) adalah suatu serangan dengan menggunakan mekanisme *injection* pada aplikasi web dengan memanfaatkan metode HTTP GET atau HTTP POST. *Cross Site Scripting* biasa digunakan oleh pihak-pihak yang berniat tidak baik dalam upaya mengacaukan konten website dengan memasukkan naskah program sebagai bagian dari teks masukan melalui formulir yang tersedia. XSS memanfaatkan lubang kelemahan keamanan yang terjadi pada penggunaan teknologi *dynamic page*. Serangan jenis ini dapat diakibatkan oleh kelemahan yang terjadi akibat ketidakmampuan server dalam memvalidasi input yang diberikan oleh pengguna. Hal ini memungkinkan halaman yang dihasilkan menyertakan perintah yang sebenarnya tidak diperbolehkan.

3. Cross Site *Request* Forgery (CSRF)

   Cross Site *Request* Forgery adalah salah satu vulnerability pada Web Application yang bekerja dengan cara mengeksploitasi suatu task dari sebuah web dengan memanfaatkan Autentikasi yang dimiliki oleh korban. CSRF juga dikenal dengan sebutan "one link" *attack*, karena pada implementasinya, sang attacker hanya butuh meng-*inject*-kan suatu link yang berisi suatu web task URL pada halaman tertentu untuk dibuka oleh calon korban, agar ketika si korban membuka halaman tersebut, secara otomatis si korban akan mengeksekusi link URL yang telah di-*inject*- kan sang attacker sebelumnya. web application yang vulnerable terhadap CSRF adalah web application yang memiliki suatu URL *Request* (baik dengan menggunakan method GET maupun POST) yang statis.

4. Web Defacement

   Serangan dengan tujuan utama merubah tampilah sebuah website – baik halaman utama maupun halaman lain terkait dengannya – diistilahkan sebagai

"Web Defacement". Hal ini biasa dilakukan oleh para "*attacker*" atau penyerang karena merasa tidak puas atau tidak suka kepada individu, kelompok, atau entitas tertentu sehingga website yang terkait dengannya menjadi sasaran utama. Pada dasarnya *deface* dapat dibagi menjadi dua jenis berdasarkan dampak pada halaman situs yang terkena serangan terkait. Jenis pertama adalah suatu serangan dimana penyerang mengubah satu halaman penuh tampilan depan. Untuk melakukan hal tersebut biasanya seorang 'defacer' harus berhubungan secara 'langsung' dengan mesin computer terkait. Hal ini hanya dapat dilakukan apabila yang bersangkutan sanggup mendapatkan hak akses penuh terhadap mesin, baik itu "root account" atau sebagainya yang memungkinkan *defacer* dapat secara interaktif mengendalikan seluruh direktori terkait. Hal ini umumnya dimungkinkan terjadi dengan memanfaatkan kelemahan pada sejumlah "services" yang berjalan disistem komputer. Jenis kedua adalah suatu serangan dimana penyerang hanya mengubah sebagian atau hanya menambahi halaman yang di-deface. Artinya yang bersangkutan mendeface suatu situs tidak secara penuh, bisa hanya dengan menampilkan beberapa kata, gambar atau penambahan "*script*" yang mengganggu. dampaknya biasanya adalah menghasilkan tampilan yang kacau atau mengganggu.

5. Phising

Phishing merupakan sebuah proses "pra-serangan" atau kerap dikatakan sebagai "*soft attack*" dimana sang penyerang berusaha mendapatkan informasi rahasia dari target dengan cara menyamar menjadi pihak yang dapat dipercaya atau seolah-olah merupakan pihak yang sesungguhnya. Contohnya adalah sebuah email yang berisi suatu informasi yang mengatakan bahwa sang pengirim adalah dari Divisi Teknologi Informasi yang sedang melakukan "*upgrading*" *system* dimana untuk memperlancar tugasnya, sang penerima email diminta untuk segera mengirimkan kata kunci "password" dari "user name" yang dimilikinya. Atau situs sebuah bank palsu yang memiliki tampilan sama persis dengan situs aslinya namun memiliki alamat URL yang

mirip, sehingga diharapkan sang nasabah akan khilaf dan secara tidak sadar memasukkan kata kunci rahasianya untuk mengakses rekening yang dimaksud. Serangan “phishing” ini kerap dikategorikan sebagai sebuah usaha “social engineering”, yaitu memanfaatkan pendekatan sosial dalam usahanya untuk mendapatkan informasi rahasia sebagai alat untuk melakukan penyerangan dikemudian hari. Modus operandi yang paling banyak ditemui saat ini adalah usaha phishing melalui SMS pada telepon genggam, dimana sudah banyak korban yang harus kehilangan uangnya karena diminta untuk melakukan transfer ke rekening tertentu dengan berbagai alasan yang seolah-olah masuk akal sehingga berhasil menjebak sang korban.

2.2.4.4. Prinsip Pengamanan Sistem

1. Otentifikasi Pemakai [2]

   Identifikasi pemakai saat login merupakan dasar asumsi sistem proteksi sehingga metode otentifikasi didasarkan pada tiga cara, yaitu sesuatu yang diketahui pemakai, yang dimiliki pemakai, dan mengenai pemakai.

2. Password

   Password merupakan salah satu otentifikasi yang diketahui pemakai, dimana pemakai memilih suatu kata-kode, mengingatnya dan mengetikkannya saat akan mengakses sistem komputer. Teknik pengamanan dengan password mempunyai beberapa kelemahan, terutama karena pemakai sering memilih password yang mudah diingatnya. Upaya untuk mengamankan proteksi password tersebut antara lain:

   A. Salting
   B. One-time Password
   C. Satu daftar pertanyaan dan jawaban yang panjang
   D. Tanggapan-tanggapan

3. Identifikasi Fisik

   Pendekatan identifikasi fisik ini dilakukan dengan memeriksa apa yang dimiliki pemakai.

A. Kartu berpita magnetic

   Kartu pengenal dengan selarik pita magnetik umumnya dikombinasikan dengan password. User akan dapat login ke komputer bila memenuhi syarat, yaitu mempunyai kartu dan mengetahui password khusus untuk kartu tersebut.

B. Sidik fisik

   Identifikasi fisik sidik jari atau sidik suara, analisis panjang jari, dan sebagainya.

C. Analisis tanda tangan

   Dengan menggunakan pena khusus, pemakai diharuskan untuk membuat tanda tangan. Dalam hal ini yang dibandingkan adalah arah gerakan dan tekanan pena saat user membuat tanda tangan.

2.2.5. SQL Injection

SQL Injection adalah metode ancaman yang mengizinkan *client* untuk mengeksekusi database melalui URL dan mendapatkan akses untuk memperoleh informasi penting. Mekanisme ancaman dilakukan dengan memanfaatkan kesalahan pada kode program yang tidak di*filter*, sehingga menyebabkan terjadinya eksploitasi pada database.[4]

2.2.5.1 Penyebab SQL Injection

Kerentanan SQL Injection paling sering terjadi ketika pengembang aplikasi web tidak menjamin bahwa nilai-nilai yang diterima dari *web form*, *cookie*, *parameter input*, dan sebagainya divalidasi atau dikodekan sebelum di*submit* untuk query SQL yang akan dieksekusi pada server database. Seorang penyerang dapat mengontrol masukan yang dikirimkan ke query SQL dan memanipulasi input, sehingga data tersebut ditafsirkan sebagai kode bukan sebagai data, dan mungkin dapat mengeksekusi kode pada database. Tanpa pemahaman yang baik dari database dan kesadaran akan isu-isu keamanan potensial dari kode yang sedang dikembangkan, pengembang aplikasi sering bisa menghasilkan aplikasi tidak aman

yang rentan terhadap SQL Injection. Berikut ini adalah tipe-tipe kesalahan dalam SQL Injection:[4]

1. *Incorrectly Handled Escape Characters.*

   Database SQL mengartikan karakter *quote* (') sebagai batas antara kode dan data. Ia diasumsikan bahwa apapun yang mengikuti karakter *quote*, maka itu akan dianggap kode yang kemudian dijalankan. Oleh karena itu, *attacker* dengan sangat mudah untuk mengetahui apakah web tersebut mempunyai celah, dengan menyisipkan karakter quote (') pada URL.

2. *Incorrectly Handled Types.*

   Karakter *quote* (‘) diinterpretasikan sebagai *string* pemisah dan digunakan sebagai batas antara kode dan data. Ketika berhadapan dengan data *numeric*, maka data tersebut tidak perlu di-enkapsulasi dengan tanda kutip. Apabila data tersebut di-enkapsulasi, maka data *numeric* akan dianggap sebagai *string*.

3. *Incorrectly Handled Query Assembly.*

   Terdapat metode *Request* pada *query* sehingga menyebabkan *attacker* dapat memanipulasi permintaan melalui HTTP dengan memisahkan nilai-nilai sebagai parameter inputan.

4. *Incorrectly Handled Errors.*

   Masalah ini terjadi ketika pesan kesalahan internal seperti database ditampilkan di *browser*. Pesan kesalahan tersebut memberikan informasi bagi *attacker* tentang sistem yang komponen yang terdapat pada sistem.

5. *Incorrectly Handled Multiple Submissions.*

   Masalah terjadi pada *web developer* yang cenderung meremehkan pengguna dalam merancang aplikasi. Misalnya, *web developer* berharap bahwa pengguna harus melewati proses 1 untuk mencapai proses 2, kemudian harus melewati proses 2 untuk mencapai proses 3. Namun demikian halnya tidak sedikit pengguna dapat mencapai proses 3 tanpa melewati proses 2.

6. *Insecure Database Configuration.*

Ketika database yang baru diinstall, biasanya mengikutsertakan *user* dan *password default*. Apabila *user* dan *password default* tersebut tidak segera diganti, maka sistem dapat diambil alih oleh orang lain.

2.2.5.2 Pencegahan SQL Injection

Langkah yang dapat ditempuh untuk mengurangi penyusupan ke halaman web dengan SQL Injection dengan cara:[5]

1. Memfilter dengan tidak membolehkan karakter seperti single quote, double quote, slash, back slash, semi colon, extended character, seperti NULL, carry return, new line, etc, dalam string form:
   a. Masukan dari from users
   b. Parameters di URL
   c. Nilai dari cookie
2. Untuk nilai numeric, convert dulu sebelum melewati statement SQL dengan mengunakan ISNUMERIC untuk meyakinkan itu adalah integer.
3. Mengubah "Startup and run SQL Server" menggunakan low privilege user dalam SQL Server Security tab.
4. Ubah stored procedure – store procedure yang tidak terpakai, seperti: *master..Xp_cmdshell, xp_startmail, xp_sendmail, sp_makewebtask.*

2.2.6 Cross Site Scripting

Cross Site Scripting adalah metode ancaman yang memaksa situs web untuk menampilkan kode berbahaya, yang dijalankan pada browser web pengguna. Mekanisme ancaman dilakukan dengan memanfaatkan kesalahan pada kode program yang tidak di*filter*. Kode tersebut akan dimasukkan ke dalam *webform*, diantaranya *form* buku tamu atau *form* pencarian.[4]

2.2.6.1 Penyebab Cross Site Scripting

Cross Site Scripting terjadi karena tidak adanya penangan nilai masukan pada variabel sehingga masukan tersebut dieksekusi begitu saja sesuai nilai yang ada. Apabila variabel tersebut tidak di *filter*, maka semua nilai masukan akan dieksekusi.

2.2.6.2 Pencegahan Cross Scripting

Langkah yang dapat ditempuh untuk mencegah ancaman *Cross Site Scripting* adalah:

1. Teknologi Static Web Pages

   Cara terbaik dan efektif untuk menghindari ancaman *Cross Site Scripting* adalah menghindari penggunaan teknologi *dynamic web page*. Halaman yang statis tentu saja memberikan kontrol yang lebih disisi server dibandingkan dengan halaman web yang dinamis. Halaman web yang dihasilkan secara statis akan memberikan kelakukan yang lebih pasti dibandingkan halaman web yang dihasilkan secara dinamis. Konsekuensi yang ditanggung adalah penyedia layanan harus merelakan sifat interaktif yang mungkin diinginkan.

2. Metode *POST*

   Metode *POST* adalah metode pengirimana data dimana variabel yang dikirimkan tidak disertakan pada link yang digunakan. Metode *POST* menyembunyikan variabel yang dikirimkan dari pengguna. Metode ini menjamin kode tidak dapat diinjeksikan melalui link yang telah didesain oleh penyerang. Link merupakan satu satunya cara yang dapat digunakan oleh penyerang untuk mengeksploitasi *Cross Site Scripting*.

3. Pengkodean Karakter Spesial pada Link

   Untuk men-non aktifkan kode script yang diinjeksikan, memerlukan pembuatan aplikasi yang mampu mengkodekan karakter tersebut, sehingga karakter tersebut tidak dapat dimengerti oleh browser yang digunakan.

4. Hilangkan kemampuan scripting

   Cross Site Scripting disebabkan oleh keberhasilan penyerang menginjeksi kode pada halaman web yang dihasilkan. Jika kode yang diinjeksikan tersebut tidak dapat diinterpretasikan, halaman web dapat ditampilkan dengan aman.

5. HTTP-Only Cookie

   Metode ini membatasi akses yang dapat dilakukan terhadap *cookie*. Dengan menggunakan *HTTP-only cookie*, browser pengguna masih dapat menerima *cookie* yang dikirimkan oleh penyedia layanan. Namun *cookie* tidak dapat

diakses melalui script yang dieksekusi pada browser pengguna. jadi script yang diinjeksikan kepada browser pengguna tidak akan dapat melakukan transfer cookie yang ada.

2.2.7. Validating Input

Validating Input adalah proses pengujian masukan yang diterima oleh aplikasi terhadap standar yang telah didefinisikan. Ada dua pendekatan yang bisa digunakan yaitu *Whitelist Validation* dan *Blacklist Validation*.[4]

2.2.7.1. Whitelist Validation

Whitelist Validation biasa disebut juga dengan *Positive Validation* adalah metode yang hanya menerima masukan yang tidak berbahaya bagi sistem. Metode ini bisa melibatkan panjang, ukuran, maupun tipe data sebelum diterima oleh sistem.

2.2.7.2 Blacklist Validation

Blacklist Validation biasa disebut juga dengan *Negative Validation* adalah metode yang menolak masukan yang berbahaya bagi sistem. Pada umumnya metode ini melibatkan masukan yang mengandung karakter, atau pola yang dianggap bisa membahayakan sistem.

2.2.8. *Firewall*

Firewall adalah suatu sistem perangkat lunak yang mengizinkan lalu lintas jaringan yang dianggap aman untuk bisa melaluinya dan mencegah lalu lintas jaringan yang dianggap tidak aman. Umumnya, sebuah *firewall* diterapkan dalam sebuah mesin terdedikasi, yang berjalan pada *gateway* antara jaringan lokal dengan jaringan Internet.

Firewall digunakan untuk membatasi atau mengontrol akses terhadap siapa saja yang memiliki akses terhadap jaringan pribadi dari pihak luar. Saat ini, istilah *firewall* menjadi istilah lazim yang merujuk pada sistem yang mengatur

komunikasi antar dua macam jaringan yang berbeda. Mengingat saat ini banyak perusahaan yang memiliki akses ke Internet dan juga tentu saja jaringan berbadan hukum didalamnya, maka perlindungan terhadap perangkat digital perusahaan tersebut dari serangan para peretas, pemata-mata, ataupun pencuri data lainnya, menjadi kenyataan.

2.2.8.1. Jenis-Jenis *Firewall*

Firewall terbagi menjadi dua jenis, yakni sebagai berikut:

1. **Personal *Firewall***

   Firewall yang didesain untuk melindungi sebuah komputer yang terhubung ke jaringan dari akses yang tidak dikehendaki. *Firewall* jenis ini akhir-akhir ini berevolusi menjadi sebuah kumpulan program yang bertujuan untuk mengamankan komputer secara total, dengan ditambahkannya beberapa fitur pengaman tambahan semacam perangkat proteksi terhadap virus, anti-spyware, anti-spam, dan lainnya. Bahkan beberapa produk *firewall* lainnya dilengkapi dengan fungsi pendeteksian gangguan keamanan jaringan (Intrusion Detection System).

2. **Network *Firewall***

   Firewall yang didesain untuk melindungi jaringan secara keseluruhan dari berbagai serangan. Umumnya dijumpai dalam dua bentuk, yakni sebuah perangkat terdedikasi atau sebagai sebuah perangkat lunak yang diinstalasikan dalam sebuah server.

2.2.8.2 Cara Kerja *Firewall*

Firewall berada diantara kedua jaringan seperti internet dan komputer sehingga *firewall* berfungsi sebagai pelindung. Tujuan utama adanya *firewall*

adalah untuk user yang tidak menginginkan lalu lintas jaringan yang berusaha masuk ke komputer, namun tidak hanya itu saja yang bisa dilakukan *firewall*. *Firewall* juga dapat menganalisis jaringan yang mencoba masuk ke komputer anda, dan dapat melakukan apa yang harus dilakukan ketika jaringan tersebut masuk. Contohnya saja, *firewall* bisa diatur untuk memblokir beberapa jenis jaringan yang mencoba keluar atau mencatat log lalu lintas jaringan yang mencurigakan.

Firewall bisa memiliki berbagai aturan yang dapat anda tambahkan atau hapus untuk menolak jaringan tertentu. Contohnya saja, hanya dapat mengakses alamat IP tertentu atau mengumpulkan semua akses dari tempat lain untuk ke satu tempat yang aman terlebih dahulu.

Firewall menurut cara kerjanya dapat dibagi menjadi tujuh jenis:

1. Packet-Filter *Firewall*

   Pada bentuknya yang paling sederhana, sebuah *firewall* adalah sebuah router atau komputer yang dilengkapi dengan dua buah NIC (Network Interface Card, kartu antarmuka jaringan) yang mampu melakukan penapisan atau penyaringan terhadap paket-paket yang masuk. Perangkat jenis ini umumnya disebut dengan *packet-filtering router*.

   Firewall jenis ini bekerja dengan cara membandingkan alamat sumber dari paket-paket tersebut dengan kebijakan pengontrolan akses yang terdaftar dalam Access Control List *firewall*, router tersebut akan mencoba memutuskan apakah hendak meneruskan paket yang masuk tersebut ke tujuannya atau menghentikannya. Pada bentuk yang lebih sederhana lagi, *firewall* hanya melakukan pengujian terhadap alamat IP atau nama domain yang menjadi sumber paket dan akan menentukan apakah hendak meneruskan atau menolak paket tersebut. Meskipun demikian, packet-filtering router

tidak dapat digunakan untuk memberikan akses (atau menolaknya) dengan menggunakan basis hak-hak yang dimiliki oleh pengguna.

Packet-filtering router juga dapat dikonfigurasikan agar menghentikan beberapa jenis lalu lintas jaringan dan tentu saja mengizinkannya. Umumnya, hal ini dilakukan dengan mengaktifkan/menonaktifkan port TCP/IP dalam sistem *firewall* tersebut. Sebagai contoh, port 25 yang digunakan oleh Protokol SMTP(*Simple Mail Transfer Protocol*) umumnya dibiarkan terbuka oleh beberapa *firewall* untuk mengizinkan surat elektronik dari Internet masuk ke dalam jaringan privat, sementara port lainnya seperti port 23 yang digunakan oleh Protokol Telnet dapat dinonaktifkan untuk mencegah pengguna Internet untuk mengakses layanan yang terdapat dalam jaringan privat tersebut. *Firewall* juga dapat memberikan semacam pengecualian (*exception*) agar beberapa aplikasi dapat melewati *firewall* tersebut. Dengan menggunakan pendekatan ini, keamanan akan lebih kuat tapi memiliki kelemahan yang signifikan yakni kerumitan konfigurasi terhadap *firewall*: daftar Access Control List *firewall* akan membesar seiring dengan banyaknya alamat IP, nama domain, atau port yang dimasukkan ke dalamnya, selain tentunya juga exception yang diberlakukan.

2. Circuit Level Gateway

Firewall jenis lainnya adalah Circuit-Level Gateway, yang umumnya berupa komponen dalam sebuah proxy server. *Firewall* jenis ini beroperasi pada level yang lebih tinggi dalam model referensi tujuh lapis OSI (bekerja pada lapisan sesi/session layer) daripada Packet Filter *Firewall*. Modifikasi ini membuat *firewall* jenis ini berguna dalam rangka menyembunyikan informasi mengenai jaringan terproteksi, meskipun *firewall* ini tidak melakukan

penyaringan terhadap paket-paket individual yang mengalir dalam koneksi.

Dengan menggunakan *firewall* jenis ini, koneksi yang terjadi antara pengguna dan jaringan pun disembunyikan dari pengguna. Pengguna akan dihadapkan secara langsung dengan *firewall* pada saat proses pembuatan koneksi dan *firewall* pun akan membentuk koneksi dengan sumber daya jaringan yang hendak diakses oleh pengguna setelah mengubah alamat IP dari paket yang ditransmisikan oleh dua belah pihak. Hal ini mengakibatkan terjadinya sebuah sirkuit virtual (*virtual circuit*) antara pengguna dan sumber daya jaringan yang ia akses.

Firewall ini dianggap lebih aman dibandingkan dengan Packet-Filtering *Firewall*, karena pengguna eksternal tidak dapat melihat alamat IP jaringan internal dalam paket-paket yang ia terima, melainkan alamat IP dari *firewall*.

3. Application Level *Firewall*

Firewall jenis lainnya adalah Application Level Gateway (atau Application-Level *Firewall* atau sering juga disebut sebagai Proxy *Firewall*), yang umumnya juga merupakan komponen dari sebuah proxy server. *Firewall* ini tidak mengizinkan paket yang datang untuk melewati *firewall* secara langsung. Tetapi, aplikasi proxy yang berjalan dalam komputer yang menjalankan *firewall* akan meneruskan permintaan tersebut kepada layanan yang tersedia dalam jaringan privat dan kemudian meneruskan respons dari permintaan tersebut kepada komputer yang membuat permintaan pertama kali yang terletak dalam jaringan publik yang tidak aman.

Umumnya, *firewall* jenis ini akan melakukan autentikasi terlebih dahulu terhadap pengguna sebelum mengizinkan pengguna

tersebut untuk mengakses jaringan. Selain itu, *firewall* ini juga mengimplementasikan mekanisme auditing dan pencatatan (logging) sebagai bagian dari kebijakan keamanan yang diterapkannya. Application Level *Firewall* juga umumnya mengharuskan beberapa konfigurasi yang diberlakukan pada pengguna untuk mengizinkan mesin klien agar dapat berfungsi. Sebagai contoh, jika sebuah proxy FTP dikonfigurasikan diatas sebuah application layer gateway, proxy tersebut dapat dikonfigurasikan untuk mengizinlan beberapa perintah FTP, dan menolak beberapa perintah lainnya. Jenis ini paling sering diimplementasikan pada proxy SMTP sehingga SMTP dapat menerima surat elektronik dari luar (tanpa menampakkan alamat e-mail internal), lalu meneruskan e-mail tersebut kepada e-mail server dalam jaringan. Tetapi, karena adanya pemrosesan yang lebih rumit, *firewall* jenis ini mengharuskan komputer yang dikonfigurasikan sebagai application gateway memiliki spesifikasi yang tinggi, dan tentu saja jauh lebih lambat dibandingkan dengan packet-filter *firewall*.

4. NAT *Firewall*

NAT (Network Address Translation) *Firewall* secara otomatis menyediakan proteksi terhadap sistem yang berada dibalik *firewall* karena NAT *Firewall* hanya mengizinkan koneksi yang datang dari komputer-komputer yang berada dibalik *firewall*. Tujuan dari NAT adalah untuk melakukan *multiplexing* terhadap lalu lintas dari jaringan internal untuk kemudian menyampaikannya kepada jaringan yang lebih luas (MAN, WAN atau Internet) seolah-olah paket tersebut datang dari sebuah alamat IP atau beberapa alamat IP. NAT *Firewall* membuat tabel dalam memori yang mengandung informasi mengenai koneksi yang dilihat oleh *firewall*. Tabel ini akan memetakan alamat jaringan internal ke alamat eksternal. Kemampuan

untuk menaruh keseluruhan jaringan dibelakang sebuah alamat IP didasarkan terhadap pemetaan terhadap port-port dalam NAT *firewall*.

5. *Stateful Firewall*

Stateful Firewall merupakan sebuah *firewall* yang menggabungkan keunggulan yang ditawarkan oleh packet-filtering *firewall*, NAT *Firewall, Circuit-Level Firewall* dan Proxy *Firewall* dalam satu sistem. *Stateful Firewall* dapat melakukan filtering terhadap lalu lintas berdasarkan karakteristik paket, seperti halnya packet-filtering *firewall*, dan juga memiliki pengecekan terhadap sesi koneksi untuk meyakinkan bahwa sesi koneksi yang terbentuk tersebut diizinlan. Tidak seperti Proxy *Firewall* atau *Circuit Level Firewall*, Stateful *Firewall* umumnya didesain agar lebih transparan (seperti halnya *packet-filtering firewall* atau NAT *firewall*). Tetapi, stateful *firewall* juga mencakup beberapa aspek yang dimiliki oleh application level *firewall*, sebab ia juga melakukan inspeksi terhadap data yang datang dari lapisan aplikasi (application layer) dengan menggunakan layanan tertentu. *Firewall* ini hanya tersedia pada beberapa *firewall* kelas atas, semacam Cisco PIX. Karena menggabungkan keunggulan jenis-jenis *firewall* lainnya, stateful *firewall* menjadi lebih kompleks.

6. Virtual *Firewall*

Virtual *Firewall* adalah sebutan untuk beberapa *firewall* logis yang berada dalam sebuah perangkat fisik (komputer atau perangkat *firewall* lainnya). Pengaturan ini mengizinkan beberapa jaringan agar dapat diproteksi oleh sebuah *firewall* yang unik yang menjalankan kebijakan keamanan yang juga unik, cukup dengan menggunakan satu buah perangkat. Dengan menggunakan *firewall* jenis ini, sebuah ISP (Internet Service Provider) dapat menyediakan layanan *firewall*

kepada para pelanggannya, sehingga mengamankan lalu lintas jaringan tersebut, hanya dengan menggunakan satu buah perangkat. Hal ini jelas merupakan penghematan biaya yang signifikan, meski *firewall* jenis ini hanya tersedia pada *firewall* kelas atas.

7. *Transparant Firewall*

Transparent *firewall* (juga dikenal sebagai *bridging firewall*) bukanlah sebuah *firewall* yang murni, tetapi ia hanya berupa turunan dari stateful *firewall*. Daripada *firewall-firewall* lainnya yang beroperasi pada lapisan IP ke atas, transparent *firewall* bekerja pada lapisan Data-Link Layer, dan kemudian ia memantau lapisan-lapisan yang ada diatasnya. Selain itu, transparent *firewall* juga dapat melakukan apa yang dapat dilakukan oleh *packet-filtering firewall*, seperti halnya stateful *firewall* dan tidak terlihat oleh pengguna (karena itulah, ia disebut sebagai Transparent *Firewall*).

2.2.9. Web Server

Web-server merupakan perangkat lunak yang melayani permintaan HTTP dari web-browser dan mengirimkan kode-kode dinamis ke server plikasi. Server aplikasi inilah yang menerjemahkan dan memproses kode-kode dinamis menjadi kode-kode statis HTML dalam suatu halaman statis yang kemudian dikirimkan ke browser oleh web-server. Web-server biasanya disebut juga sebagai HTTP server karena basisnya menggunakan protokol HTTP. Ketika user mengetik sebuah URL atau meng-klik sebuah link, maka web-browser akan mengirimkan permintaan halaman web. Melalui alamat URL, diketahui server mana yang menyimpan halaman web tersebut, selanjutnya server menerima permintaan dan mencari halaman web yang diminta, apabila halaman web yang diminta terdapat diwebserver tersebut, maka server akan mengirimkan alaman web yang diminta melalui Jaringan Internet atau Intranet ke browser di komputer klien. Ketika halaman tersebut tiba di komputer, web-browser

segera menerjemahkan bahasa hypertext dan menampilkannnya di komputer klien, proses ini hanya berjalan dalam hitungan detik. komputer klien dapat mengakses homepage dan halaman web secara online apabila terkoneksi ke suatu jaringan komputer, sebab web server selalu dikoneksikan melalui jaringan komputer. Melalui jaringan inilah komputer klien dapat berinteraksi langsung dengan jutaan web-server yang tersebar di dunia, begitu juga dengan web-server yang akan melayani ribuan bahkan jutaan permintaan halaman web setiap harinya.

Banyak sekali aplikasi Web Server saat ini baik yang berbayar ataupun yang gratis. Diantara web server yang gratis adalah :

1. Apache
2. Lighttpd
3. Internet Information Services (IIS)
4. Litespeed
5. Nginx
6. Cherokee
7. Hiawatha

Web Server Apache

Apache merupakan *web server* yang paling banyak dipergunakan di internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya Apache mengeluarkan programnya yang dapat dijalankan di windows NT. Berdasarkan sejarahnya, Apache dimulai oleh veteran developer NCSA httpd (*National Center for Supercomputing Application*). Saat itu pengembangan NCSA httpd sebagai web server mengalami stagnasi. ROB MC COOL meninggalkan NCSA dan memulai sebuah proyek baru bersama para webmaster lainnya, menambal bug, dan menambahkan fitur pada NCSA httpd. NCSA mengembangkan program ini lewat mailing list. Dengan berpijak pada NCSA httpd versi 1.3, Team Apache mengeluarkan rilis pertama kali secara resmi Apache versi 0.6.2. Tim inti pengembang Apache waktu itu :

1. Brian Behlendorf
2. Roy T. Fielding
3. Rob Hartill
4. David Robinson
5. Cliff Skolnick
6. Randy Terbush
7. Robert S. Thau
8. Andrew Wilson

Dengan tambahan kontribusi dari :

1. Eric Hagberg
2. Frank Peters
3. Nicolas Pioch

Nama Apache diambil dari kata "*A Patchy Server*", server perbaikan yang penuh dengan tambalan *(patch)*. Tambalan yang dimaksud adalah penambahan fitur dan penambalan bug dari NCSA httpd Versi 1.3. Saat ini Apache dipergunakan secara luas. Hal ini disebabkan karena programnya yang gratis, dengan kinerja relatif stabil. Dalam pengembangannya pun mempergunakan sistem Bazaar, yakni tiap orang dibuka kesempatan seluas-luasnya untuk dapat memberikan kontribusi dalam mengembangkan program. Apache mempunyai program pendukung yang cukup banyak.

Web Server Litespeed

Litespeed web server merupakan teknologi baru yang diciptakan untuk menggantikan Apache Web Server di masa mendatang. Litespeed menurut hasil riset mempunyai kelebihan yaitu 50% lebih cepat dalam memproses PHP dan lebih cepat 6x dibandingkan Apache Web Server.

Web Server Nginx

Nginx dengan cepat memberikan konten statis dengan penggunaan efisien sumber daya sistem.Hal ini dapat menyebarkan dinamis HTTP konten dijaringan

menggunakan FastCGI *handler* untuk *script*, dan dapat berfungsi sebagai perangkat lunak yang sangat mampu penyeimbang beban. Nginx menggunakan asynchronous - event pendekatan untuk menangani permintaan yang diprediksi memberikan kinerja yang lebih bawah beban, kontras dengan Apache HTTP server model yang menggunakan berulir atau proses yang berorientasi pada pendekatan-permintaan penanganan.

Web Server IIS

Internet Information Service (IIS) adalah komponen yang dapat digunakan untuk mengelola web, *File Transfer Protocol* (FTP), Ghoper, dan NNTP. Komponen IIS terdapat pada sistem operasi Windows NT, Windows 2000, Windows XP, 7 dan Windows Server 2003. Pada Windows 98, supaya dapat mengelola Web diperlukan komponen Personal Web Server (PWS) walaupun tidak terdapat fasilitas untuk FTP. PWS juga merupakan bagian dari IIS. Komponen PWS terdapat pada CD master Windows 98 dan terletak pada direktori \add-ons\pws. Instalasi PWS dapat dilakukan dari <Drive CD ROOM>:\ADD-ONS|PWS|SETUP.