WEBSITE CONTROL PADA JARINGAN UNTUK KEAMANAN DAN KENYAMANAN BAGI BROWSER DENGAN METODE JAVA SERVLET

ABSTRAK

Kemudahan bertransaksi dalam pengiriman atau penerimaan data lewat internet menarik minat semua user, baik yang memiliki akses ke dalam jaringan maupun yang tidak memiliki akses seperti hacker/cracker. Internet adalah jaringan komputer raksasa yang terhubung dan dapat saling berinteraksi berkat pesatnya perkembangan teknologi jaringan. Tetapi dalam beberapa hal terhubung dengan internet bisa menjadi suatu ancaman. Banyak serangan yang dapat terjadi baik dari dalam maupun luar seperti virus, trojan, maupun hacker. Dalam hal ini security komputer dan jaringan komputer berperan penting. SQL Injection yang biasa digunakan selama ini tidak relevan, karena masih terdapat kebocoran data dalam jaringan. Dalam penulisan ini digunakan Metode Java Servlet yang mempunyai tingkat keamanan cukup tinggi untuk meminimalisir kebocoran data karena menggunakan pemrograman berbasis pada OOP (Object Oriented Programming).

Kata Kunci : Jaringan, Security, SQL, Hacker, Servlet

M.S. Herawati

Jurusan Sistem Informasi,
Fakultas Ilmu Komputer
Universitas Gunadarma
msherawati@staff.gunadarma.ac.id

PENDAHULUAN

Website merupakan salah satu *killer applications* yang menyebabkan populernya internet. Kehebatan website adalah kemudahannya untuk mengakses informasi, yang dihubungkan satu dengan lainnya melalui konsep *hypertext*. Informasi dapat tersebar di mana-mana dan terhubung melalui *hyperlink*. Informasi lebih lengkap tentang website dapat diperoleh di web W3C.

Pembaca atau peraga sistem website yang lebih dikenal dengan istilah *browser*Êdapat diperoleh dengan mudah, murah atau gratis. Contoh browser adalah Netscape, Internet Explorer, Opera, kfm (KDE file manager di sistem Linux), dan masih banyak lagi.

Kemudahan penggunaan program *browser* inilah yang memicu populernya Website. Sejarah browser dimulai dari *browser* di sistem komputer NeXT yang kebetulan digunakan oleh Berners-Lee. Selain *browser* NeXT itu, pada saat itu baru ada browser yang berbentuk teks *(text-oriented)* seperti *line mode browser*. Berkembangnya website dan internet menyebabkan pergerakan sistem informasi untuk menggunakannya sebagai basis.

Banyak sistem tidak terhubung ke internet tetapi tetap menggunakan website sebagai basis untuk sistem informasinya yang dipasang di jaringan intranet. Keamanan sistem informasi berbasis website dan teknologi internet bergantung kepada keamanan sistem website. Arsitektur sistem website terdiri dari dua sisi, yakni server dan client. Keduanya dihubungkan dengan jaringan komputer *(computer network)*. Selain menyajikan data dalam bentuk statis, sistem website dapat menyajikan data dalam bentuk dinamis dengan menjalankan program yang dijalankan di server (misal dengan CGI, servlet) dan client (applet, Javascript).

Munculnya masalah keamanan jaringan didasarkan atas beberapa asumsi dari pihak user, web master, dan sistem web itu sendiri. Asumsi dari pihak user (pengguna) adalah sebagai berikut:

1. Server dimiliki dan dikendalikan oleh organisasi yang mengaku memiliki server.
2. Dokumen yang ditampilkan bebas dari virus, *trojan horse*, atau itikad jahat lainnya. Bisa
   saja seorang yang nakal memasang virus di webnya, tapi ini suatu anomali.
3. Server tidak mendistribusikan informasi mengenai pengunjung (*user* yang melakukan
   browsing) kepada pihak lain. Hal ini disebabkan ketika user mengunjungi sebuah web site, data-data tentang dia (nomor IP, *operating system*, browser yang digunakan, dll.) dapat dicatat.
4. Pelanggaran terhadap asumsi ini melanggar privasi. Jika ini terjadi maka pengunjung tidak akan kembali ke situs itu lagi.

Asumsi dari penyedia layanan *(website master)* adalah sebagai berikut:

1. Pengguna tidak beritikad untuk merusak server atau mengubah isinya (tanpa izin).
2. Pengguna hanya mengakses dokumen-dokumen atau informasi yang diizinkan diakses.
   Seorang pengguna tidak mencoba-coba masuk ke direktori yang tidak diperkenankan
   (istilah yang umum digunakan adalah *directory traversal*).
3. Identitas pengguna benar. Banyak situs web yang membatasi akses kepada *user* tertentu.

Dalam hal ini, jika seorang pengguna "*login*" ke web, maka dia adalah pengguna yang benar.

Asumsi dari kedua belah pihak adalah bahwa jaringan komputer dan komputer bebas dari penyadapan pihak ketiga. Informasi yang disampaikan dari server ke pengguna (dan sebaliknya) terjamin keutuhannya dan tidak dimodifikasi oleh pihak ketiga. Asumsi-asumsi itu, jika dilanggar, akan menimbulkan masalah keamanan pada komputer.

Internet merupakan jaringan global yang menghubungkan satu network dengan network lain. TCP/IP menjadi protokol penghubung antara jaringan-jaringan yang beragam untuk berkomunikasi. Website merupakan bagian dari internet yang paling cepat berkembang dan paling populer

Website bekerja berdasarkan tiga mekanisme yakni protokol standar aturan, *address*, dan HTML. Protokol standar aturan digunakan untuk berkomunikasi pada *computer networking*. Hypertext Transfer Protocol (HTTP) adalah protokol untuk website. Address website mempunyai aturan penamaan alamat web yakni URL *(Uniform Resource Locator)* yang diguna-kan sebagai standar alamat internet. Sedangkan HTML digunakan untuk membuat dokumen yang diakses melalui web. HTML merupakan standar bahasa yang digunakan untuk menampilkan documen web, mengontrol tampilan dari *web page* dan *content*nya, mempubli-kasikan dokumen secara online, dan membuat *online form* yang dapat diguna-kan untuk menangani pendaftaran dan transaksi secara *online*.

Browser merupakan software yang di-*install* di mesin client yang berfungsi untuk menerjemahkan tag-tag HTML menjadi halaman web. Browser yang sering digunakan adalah Internet Explorer, Netscape Navigator, dan Opera, Mozilla.

Teknologi Java Servlet

Servlet merupakan class yang didefinisikan dalam java dan digunakan untuk meningkatkan kemampuan web server dalam menangani *request* dan *response* dari client. Servlet dapat menerima *request* dan menghasilkan *response* melalui protokol komunikasi yang berbeda, tetapi sebagian besar tipe yang digunakan adalah HTTP Servlet, yang diimplementasikan dengan class java javax.servlet.httpServlet.

Java memiliki dua paket yang menyediakan interface dan class untuk servlet, yaitu javax.servlet dan javax.servlet.http. Tabel 1 memperlihatkan beberapa metode yang terdapat pada interface servlet.

Tabel 1.
Metode Pada Interface Servlet

Metode	Keterangan
void init (ServletConfig config)	
	Metode ini dipanggil hanya sekali selama siklus servlet dan pemanggilan secara otomatis. Digunakan untuk inisialisasi servlet.
ServletConfig getServletConfig()	
	Metode ini mengembalikan reference ke objek yang mengimplementasikan interface ServletConfig. Objek ini memungkinkan servlet mengakses konfigurasi informasi servlet seperti inisialisasi parameter dan ServletContex, yang memungkinkan servlet memiliki akses ke lingkungannya, seperti pengaksesan ke server di mana servlet dijalankan.
void service (ServletRequest request, ServletResponse response)	
	Metode ini merupakan metode pertama yang dipanggil pada setiap servlet untuk merespons request dari client.
String getServletInfo()	
	Metode ini didefinisikan programmer servlet untuk mengembalikan String yang berisi informasi servlet seperti author dan versi servlet.
Void destroy()	
	Metode ini sering disebut clenUp, dan digunakan untuk mendealokasi resource yang digunakan servlet.

Paket servlet mendefinisikan dua buah class abstract yang mengimplementasikan interface Servlet, yaitu class GenericServlet pada paket javax.servlet dan class HttpServlet pada paket javax.servlet.http. Contoh-contoh yang diberikan pada bagian ini diturunkan dari class HttpServlet, yang mendefinisikan kemampuan pemrosesan servlet untuk mewarisi fungsionalitas web server.

Metode utama yang terdapat pada setiap servlet adalah metode service, yang memiliki parameter berupa objek ServletRequest dan ServletResponse. Kedua objek ini menye-diakan akses untuk *input stream* maupun *output stream* dan mengizinkan servlet untuk membaca data dari client dan mengirimkan data kembali ke client. Jika terdapat masalah selama eksekusi servlet, maka class Servlet Exception atau IOException di-*passing* untuk memberitahukan masalah yang terjadi.

Class HttpServlet melakukan *override method service* untuk membedakan antara request yang diterima web browser client. Dua tipe request yang paling umum digunakan adalah GET dan POST, yang sering disebut sebagai *request method*. Request GET digunakan untuk menerima informasi dari server yang berupa file HTML atau *image*. Sedangkan request POST digunakan untuk mengirim data ke server dalam bentuk HTML yang berisi data yang dimasukkan oleh client. Metode yang sering digunakan untuk merespon request dari client adalah doGet dan doPost yang memiliki dua parameter, yaitu HttpServletRequest dan HttpServlet Response. TAbel 2 dan 3 menyajikan beberapa metode yang berasal dari ServletRequest dan Servlet Response.

Tabel 2.
Beberapa Metode dari ServletRequest

Metode	Keterangan
String getParameter(String name)	
	Dikirim ke servlet sebagai bagian request GET atau POST.
Enumeration getParameterNames()	
	Mengembalikan nama seluruh parameter yang dikirim ke servlet sebagai bagian dari request POST.
String[] getParameterValues(String name)	
	Mengembalikan *array of string* yang berisi nilai untuk parameter servlet
Cookie[] getCookies()	
	Mengembalikan *array of cookie*, merupakan objek client yang disimpan di server. Cookies dapat digunakan untuk mengidentifikasikan secara unik setiap client oleh server.
HttpSession getSession(Boolean create)	
	Mengembalikan objek HttpSession yang sedang berlangsung pada client. Parameter akan bernilai jika tidak terdapat objek Httpsession di client. Secara fungsional memiliki kegunaan yang sama dengan Cookies.

HASIL DAN PEMBAHASAN

Keamanan Server

Server Website menyediakan fasilitas agar client dari tempat lain dapat mengambil informasi dalam bentuk berkas *(file)*, atau mengeksekusi perintah (menjalankan pro-gram) di server. Fasilitas pengambilan berkas dilakukan dengan perintah "GET", sementara mekanisme untuk mengeksekusi perintah di server dapat dilakukan dengan "CGI" *(Common Gateway Interface)*, *Server Side Include* (SSI), *Active Server Page* (ASP), PHP, atau dengan menggunakan *servlet* (seperti pernggunaan *Java Servlet*).

Kedua jenis servis di atas (mengambil berkas biasa maupun menjalankan program di server) memiliki potensi lubang keamanan yang berbeda. Adanya lubang keamanan di sistem website dapat dieksploitasi dalam bentuk yang beragam, antara lain:

1. Informasi yang ditampilkan di server diubah sehingga dapat mempermalukan perusahaan atau organisasi (dikenal dengan istilah *deface*1);
2. Informasi yang semestinya dikonsumsi untuk kalangan terbatas (misalnya laporan keuangan, strategi perusahaan, atau database client) ternyata berhasil disadap oleh pesaing (ini mungkin disebabkan salah setup server, salah setup router/ firewall, atau salah setup authentication);

Tabel 3.
Beberapa Metode dari Servlet Response

Metode	Keterangan
void addCookie(Cookie cookie)	
	Digunakan untuk menambahkan cookie pada header sebagai respons ke client.
ServletOutputStream getOutputStream()	
	Mendapatkan output stream berbasis byte yang memungkinkan data teks dikirim ke client.
PrintWriter getWriter()	
	Mendapatkan output stream berbasis karakter yang memungkinkan data binary dikirim ke client.
Void setContentType(String type)	
	Menspesifikasi tipe MIME respons ke browser.

3. Informasi dapat disadap (seperti misalnya pengiriman nomor kartu kredit untuk membeli melalui Website, atau tindakan memonitor orang yang melakukan *web surfing*).
4. Server diserang (misalnya dengan memberikan *request* secara bertubi-tubi) sehingga tidak bisa memberikan layanan ketika dibutuhkan (*denial of service attack*);
5. Untuk server web yang berada di belakang firewall, lubang keamanan di server web yang dieksploitasi dapat melemahkan atau bahkan menghilangkan fungsi dari firewall (dengan mekanisme *tunneling*).

Strategi Implementasi

Strategi implementasi mencakup tindakan membatasi akses melalui kontrol akses, proteksi halaman dengan mengunakan password, secure socket layer, mengetahui jenis server, dan keamanan program CGI.

Sebagai penyedia informasi (dalam bentuk berkas), sering diinginkan pembatasan akses. Misalnya, diinginkan agar hanya orang-orang tertentu yang dapat mengakses berkas (informasi) tertentu. Pada prinsipnya ini adalah

masalah kontrol akses. Pembatasan akses dapat dilakukan dengan, membatasi domain atau nomor IP yang dapat mengakses, menggunakan pasangan userid & password, dan mengenkripsi data sehingga hanya dapat dibuka (dekripsi) oleh orang yang memiliki kunci pembuka.

Salah satu mekanisme untuk mengatur akses adalah dengan menggunakan pasangan *userid* (*user identification*) dan *password*. Untuk server website yang berbasis Apache1, akses ke sebuah halaman (atau sekumpulan berkas yang terletak di sebuah direktori di sistem Unix) dapat diatur dengan meng-gunakan berkas ".htaccess".

Cara lain untuk meningkatkan keamanan server website adalah dengan menggunakan enkripsi pada komunikasi pada tingkat socket. Dengan menggunakan enkripsi, orang tidak bisa menyadap data-data (transaksi) yang dikirimkan dari/ke server website. Salah satu mekanisme yang cukup populer adalah dengan menggunakan *Secure Socket Layer* (SSL) yang mulanya dikembangkan oleh *Netscape*.

Informasi tentang web server yang digunakan dapat dimanfaatkan oleh perusak untuk melancarkan serangan sesuai dengan tipe server dan *operating system* yang digunakan. Seorang penyerang akan mencari tahu software dan versinya yang digunakan sebagai web server, kemudian mencari informasi di internet tentang kelemahan web server tersebut. Informasi tentang program server yang digunakan sangat mudah diperoleh. Cara yang paling mudah adalah dengan menggunakan program "telnet" dengan melakukan telnet ke port 80 dari server web tersebut, kemudian menekan tombol *return* dua kali. Web server akan mengirimkan respon dengan didahului oleh informasi tentang server yang digunakan.

Common Gateway Interface (CGI) digunakan untuk menghubungkan sistem website dengan software lain di server web. CGI memungkinkan adanya hubungan interaktif antara user dan server web. CGI seringkali digunakan sebagai mekanisme untuk memperoleh informasi dariÊuser melalui "fill out form", mengakses database, atau menghasilkan halaman yang dinamis.

Membuat Servlets dan Konfigurasi Tomcat pada Eclipse

Untuk membuat Servlet dan konfigurasi Tomcat pada Eclipse, tools-tools yang dibutuhkan antara lain:

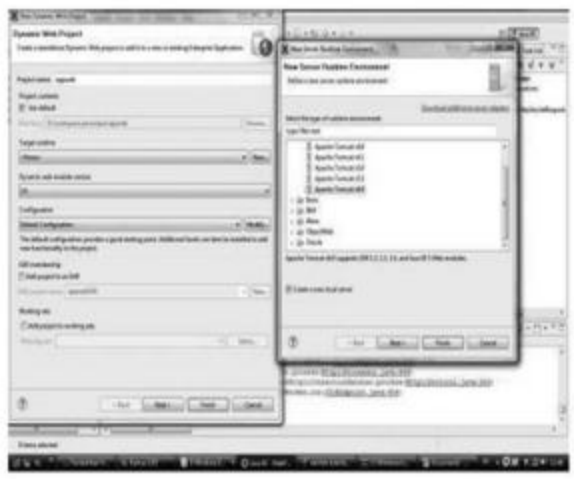
1. JEE dari Sun Microsystem. Jika belum memilikinya, silahkan ikuti link di bawah ini: http://java.sun.com/javaee/downloads/ index.jsp?userOsIndex=6&userOsId=windows&userOsName=Windows
2. Apache Tomcat sebagai web server. Jjika belum memiliki silahkan ikuti link di bawah ini: http://apache.pesat.net.id/tomcat/tomcat-6/v6.0.20/bin/apache-tomcat 6.0.20.zip
3. Eclipse sebagai editor. Jika belum memiliki silahkan ikuti link di bawah ini: http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/galileo/R/eclipse-jee-galileo-win32.zip&url=http://kambing.ui.ac.id/eclipse//technology/epp/downloads/release/galileo/R/eclipse-jee-galileo-win32.zip&mirror_id=1012

Setelah semua tools-tools di atas diinstall, aplikasi servlets pertama mulai dilakukan.

1. Buka Eclipse IDE, buat sebuah dinamic web project.

Gambar 1. Eclipse IDE

2. Beri nama projectnya, kemudian pada target runtime, klik new, pilih web server tomcat 6.0, kemudian finish.



Gambar 2. Tahapan pemberian nama Project

3. Setelah itu pada project explorer akan tampil nama project yang barusan dibuat. Sekarang kita buat sebuah kelas java dengan nama TestServlets.java. Caranya, klik pada JavaResource : src, klik kanan, pilih new class. maka akan muncul dialog.

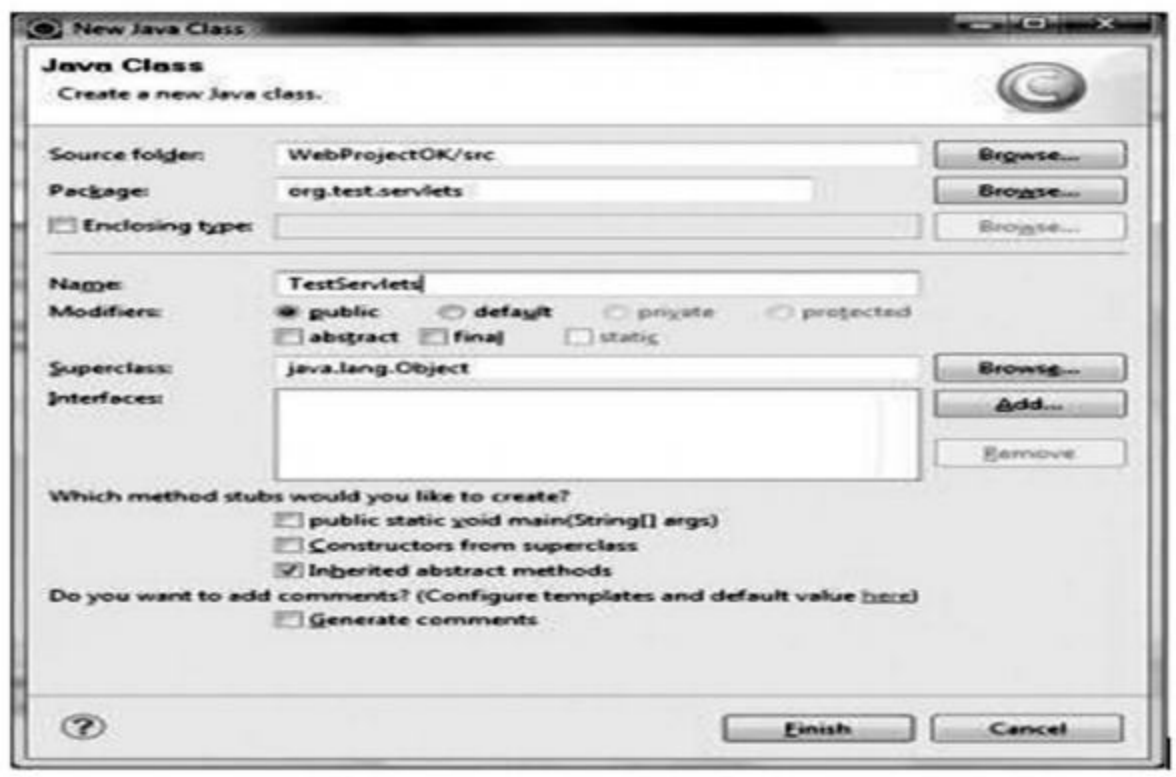
Isikan dalam package org.test.servlets untuk menempatkan dalam sebuah package. kemudian *finish*.

4. Isikan koding didalam kelas yang barusan dibuat

```
package org.test.webapp;
import java.io.IOException;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import
javax.servlet.http.HttpServletRequest;
import
javax.servlet.http.HttpServletResponse;
public class TestServelt extends
HttpServlet {
@Override
protected void doGet(HttpServletRequest
req, HttpServletResponse resp)
throws ServletException, IOException {
resp.getWriter().println(new Date());
resp.getWriter().println("EL_Zalman's
sedang beraksi");
}
}
```

Gambar 4. Editor Penulisan Program Web.xml

6. Setelah itu untuk menjalankannya, setting dulu window yang diperlukan untuk mengatur web server tomcat. masuk ke menubar window->open perspectiv->other



Gambar 3. Kotak Dialog TextServlet

Gambar 5. Tahap penyetingan Window untuk Web Server Tomcat setelah itu pilih server

Gambar 6. Tampilan show view

7. Setelah itu akan tampil sebuah jendela baru berjudul server pada kumpulan window di bawah editor koding. klik pada nama jendela bernama server, klik kanan pilih new->server

Gambar 7. Jendela Baru Server

maka akan tampil dialog, pilih tomcat 6, lalu *finish*

Gambar 8: Dialog pemilihan Tomcat 6

8. Setelah konfigurasi tomcat selesai, masukan project web dinamis dalam web server, caranya klik kanan pada localhost, pilih add remove server

Gambar 9: Project Web Dinamis

setelah itu add project kita, kemudian *finish*

9. Bila semua sudah selesai, tinggal mencoba pada browser, ketikan http://localhost:8080/appweb
10. Akan muncul tulisan yang telah kita buat pada TestServlets.java

Keamanan Client Website

Keamanan di sisi client biasanya berhubungan dengan masalah *privacy* dan penyisipan virus atau trojan horse. Ketika kita mengunjungi sebuah situs website, browser dapat "dititipi" sebuah "*cookie*" yang fungsinya untuk menandai kita. Ketika kita berkunjung ke server itu kembali, maka server dapat mengetahui bahwa kita kembali dan server dapat memberikan setup sesuai dengan keinginan (*preference*) kita. Ini merupakan servis yang baik. Namun data-data yang sama juga dapat digunakan untuk melakukan *tracking* ke mana saja kita pergi. Ada juga situs web yang mengirimkan script (misal Javascript) yang melakukan interogasi terhadap server kita (melalui browser) dan mengirimkan informasi ini ke server. Bayangkan jika di dalam komputer kita terdapat data-data yang bersifat rahasia dan informasi ini dikirimkan ke server milik orang lain.

Cara penyerangan terhadap client yang lain adalah dengan menyisipkan virus atau trojan horse. Bayangkan apabila yang anda *download* adalah virus atau trojan horse yang dapat menghapus isi harddisk anda. Salah satu contoh yang sudah terjadi adalah adanya web yang menyisipkan trojan horse Back Orifice (BO) atau Netbus sehingga komputer anda dapat dikendalikan dari jarak jauh. Orang dari jarak jauh dapat menyadap apa yang anda ketikkan, melihat isi direktori, melakukan reboot, bahkan memformat harddisk.

KESIMPULAN

Selain menyajikan data dalam bentuk statis, sistem website dapat menyajikan data dalam bentuk dinamis dengan menjalankan program. Program dapat dijalankan di server (misalnya dengan CGI, servlet) dan di client (applet, Javascript).

Adanya lubang keamanan di sistem website dapat dieksploitasi dalam bentuk yang beragam, di antaranya informasi yang ditampilkan di server diubah sehingga dapat mempermalukan perusahaan atau suatu organisasi. Informasi yang semestinya dikonsumsi untuk kalangan terbatas ternyata berhasil disadap oleh saingan. Informasi yang penting bagi perusahaan dapat disadap. Server diserang dengan memberikan *request* secara bertubi-tubi, sehingga tidak bisa memberikan layanan ketika dibutuhkan dan untuk server web yang berada di belakang firewall lubang keamanan di server web yang dieksploitasi dapat melemahkan atau bahkan menghilangkan fungsi dari firewall.

Strategi implementasi yang dapat dilakukan adalah dengan membatasi akses melalui kontrol akses, proteksi halaman dengan menggunakan *password, secure socket layer* (salah satu cara untuk meningkatkan keamanan server website adalah dengan menggunakan enkripsi pada komunikasi di tingkat socket), mengetahui jenis server yang digunakan, dan pengadaan CGI memungkinkan hubungan interaktif antara user dan server web, di mana CGI seringkali digunakan sebagai mekanisme untuk memperoleh informasi dariÊuser melalui "fill out form", mengakses database, atau menghasilkan halaman yang dinamis.

DAFTAR PUSTAKA

Baker, Richard H. 1995. *Network Security: how to plan for it and achieve it*. McGraw- Hill International: New York.

Bellovin, Steven M. "Security Problems in TCP/IP Protocol Suite." *Computer Communication Review*, Vol. 19, No. 2, pp. 32-48, 1989.

Tim Berners-Lee, Tim. 2000. *Weaving the Web: the past, present and future of the world wide web by its inventor*. Texere. www.ilmukomputer.com

https://www.google.co.id/#q=konfigurasi+java+servlet+pada+database/J*avanewbie.wordpress.com/java/java.../servlets/awal-membuat-servlets-da...ý lecturer.eepisits.edu/~idris/files/oop_lanjut/P11a_servlet.pdfý*