BAB II
TINJAUAN PUSTAKA

Pada bab dua, terdiri dari kajian pustaka dan dasar teori. Kajian pustaka adalah membahas penelitian yang telah ada dan yang diusulkan. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan. Pada penelitian ini, dasar teori yang diperlukan adalah dasar teori yang berdasarkan latar belakang dan rumusan masalah.

2.1 Kajian pustaka

Kajian pustaka membahas penelitian yang telah ada dan yang diusulkan. Penelitian sebelumnya berjudul "*Secured Authentication Protocol System using Images"*. Penelitian yang diusulkan saat ini akan diimplementasikan pada *website ecommerce* PTIIK. Perbedaan antara penelitian sebelumnya dan yang diusulkan adalah pada proses autentikasi dan pengujian yang dilakukan. Pada penelitian sebelumnya menggunakan *hidden characters* untuk melakukan proses *mapping* terhadap komponen-komponen yang ada sedangkan pada penelitian yang diusulkan menggunakan tabel yang menyimpan urutan gambar dan urutan karakter untuk proses *mapping* terhadap komponen-komponen yang ada. Perbedaan yang kedua adalah pada sisi pengujian. Penelitian sebelumnya lebih difokuskan pada pengujian keamanan sedangkan penelitian yang diusulkan akan menguji dari sisi keamanan dan dari sisi pengguna (*user acceptance testing*)

2.2 Autentikasi

Autentikasi adalah suatu mekanisme yang digunakan oleh suatu sistem untuk mengidentifikasi user yang berhak mengakses informasi pada sistem tersebut. Mekanisme autentikasi yang ada saat ini dibagi ke dalam tiga bentuk umum:

a. *Token-based system* menggunakan suatu objek yang eksklusif yang hanya dimiliki oleh *user* tertentu sebagai bentuk identifikasi. Salah satu contohnya adalah penggunaan kartu ATM, dimana setiap pemilik memiliki satu kartu untuk proses autentikasi.
b. *Biometrics-based system* menggunakan ciri khas tubuh tertentu dari *user* sebagai proses autentikasi. Cara ini memiliki tingkat keamanan yang sangat tinggi, namun belum banyak digunakan karena membutuhkan biaya

yang cukup mahal dan proses identifikasi yang cukup lama. Beberapa contohnya adalah *fingerprints*, iris *scan*, dan *facial recognition*.

c. *Knowledge-based system* menggunakan suatu informasi yang hanya diketahui oleh *user*. Sistem ini dibagi ke dalam dua kategori umum yaitu tekstual *password* dan *graphical password*.[HAR-10]

Autentikasi dapat dikatakan aman apabila terhindar dari *password attacks*. *Password attacks* yang dapat terdiri dari:

1. *Brute Force Attacks*

   Pada tipe serangan ini, semua kombinasi *password* yang mungkin dicoba untuk menemukan *password* pengguna. *Brute force attack* biasanya digunakan untuk membuka *password* yang dienkripsi dimana *password* tersebut disimpan di form teks terenkripsi.

2. *Dictionary Attack*

   Tipe serangan ini relatif lebih cepat daripada *brute force*. tidak seperti *brute force* yang mencoba semua kemungkinan, *dictionary attack* mencoba menemukan *password* dengan kata-kata yang sering digunakan untuk dijadikan *password*. Meskipun *dictionary attack* lebih cepat daripada *brute force*, *dictionary attack* ini juga mempunyai keterbatasan. *Dictionary attack* bisa saja tidak dapat menemukan *password* pengguna karena tidak terdapat pada daftar *password* tersebut

3. *Phishing Attacks*

   Phising attacks adalah serangan berbasis *web* dimana penyerang mengalihkan pengguna ke *website* palsu untuk mendapatkan *password* atau kode PIN dari pengguna tersebut.

4. *Shoulder Surfing*

   Shoulder surfing adalah salah satu alternatif memata-matai dimana penyerang mengamati pergerakan dari pengguna saat memasukkan *password*

5. *Key Loggers*

   Program ini memonitor aktivitas pengguna dengan merekam setiap tombol ditekan oleh pengguna. Penyerang menginstal *software key logger* ke dalam sistem pengguna, baik dengan menginstal sendiri atau dengan

menipu pengguna untuk mengklik agar menginstal *file* tersebut dalam sistem. *Key logger* membuat *file log* dari tombol ditekan oleh pengguna dan kemudian mengirimkan *file log* ke alamat *e-mail* penyerang. Penyerang kemudian mendapatkan *password* dan dapat mengakses ke sistem target

6. *Video Recording Attack*

   Dalam jenis serangan para penyerang dengan bantuan kamera yang dilengkapi ponsel atau kamera mini, menganalisis rekaman video dari pengguna yang memasukkan *password*.

7. *Replay Attacks*

   Replay attack adalah bagian dari *Passive Man In the Middle Attack*. *Man in the middle* pasif adalah serangan pada jaringan dimana penyerang "mendengar" percakapan antara pengirim (AP) dan penerima (*Client*) seperti mengambil sebuah informasi yang bersifat rahasia seperti pada proses autentikasi, lalu *hacker* menggunakan informasi tersebut untuk berpura-pura menjadi client yang terautentikasi [RIS-12].

Pada skripsi kali ini, penulis menggunakan *brute force* dan *man in the middle* pasif untuk pengujian keamanan autentikasi.

2.3 Tekstual Password

Tekstual *password* adalah bentuk yang paling umum dan sering digunakan saat ini. Tekstual *password* menggunakan karakter alfanumerik (ASCII) untuk mengidentifikasi *user*. *User* akan diminta untuk memasukkan kombinasi dari beberapa karakter pada proses autentikasi.

Berdasarkan studi yang dilakukan, user lebih sering menggunakan kata-kata yang pendek dan mudah diingat sebagai *password*. Sayangnya, bentuk *password* seperti ini sangat mudah ditebak. Di sisi lain, *password* yang susah ditebak cukup sulit untuk diingat. Dengan kemampuan user yang hanya mampu mengingat beberapa *password* saja, *user* cenderung mencatat semua *password* atau menggunakan *password* yang sama pada semua akun yang dimiliki. Selain masalah daya ingat manusia, saat ini juga sedang marak penggunaan *spyware* untuk menangkap informasi berupa "*username*" dan "*password*" yang diketikkan *user* untuk dikirim ke penyerang. Hal ini menandakan bahwa tekstual *password*

yang menggunakan input dari *keyboard* cukup mudah untuk diserang. Belajar dari kelemahan-kelemahan yang dimiliki textual *password*, kemudian diciptakanlah suatu metode baru yang dikenal dengan *Graphical Password*. [HAR-10]

2.4 Graphical Password

Metode *Graphical Password* pertama kali dikemukakan oleh G. Blonder pada 1996. Graphical password dianggap mampu menggantikan tekstual *password* sebagai metode autentikasi *user*. Hal ini mengacu pada penelitian psikologis bahwa gambar lebih mudah untuk dikenali dan diingat oleh memori manusia jika dibandingkan dengan teks.

Berdasarkan teknik identifikasinya, *graphical password* terbagi dua yaitu:

a. *Recognition-based password*, pada model ini *user* diminta untuk mengenali gambar yang dipilih pada awal registrasi.
b. *Recall-based password*, pada model ini *user* diminta untuk membuat kembali gambar yang telah digambar atau dipilih pada awal registrasi.

Berdasarkan tipe latar gambar yang digunakan, *graphical password* terbagi dua yaitu:

a. *Image-based password*, pada model ini *password* menggunakan gambar sebagai *background* dari *password*. Model ini biasanya menggunakan teknik identifikasi *Recognition-based*..
b. *Grid-based password*, pada model ini *password* menggunakan *grid* sebagai *background* dari *password*. Model ini biasanya menggunakan teknik identifikasi *Recall-based*. [HAR-10]

2.5 *One Time Password*

One Time Password merupakan mekanisme *login* ke dalam sebuah jaringan atau layanan dengan menggunakan *password* yang unik yang hanya dapat digunakan satu kali saja. Hal ini digunakan untuk mencegah berbagai bentuk pencurian identitas dengan memastikan bahwa kombinasi nama atau *password* pengguna tidak dapat digunakan sebanyak dua kali. *One time password* adalah bentuk dari autentikasi yang kuat dan menawarkan perlindungan yang lebih efektif untuk rekening bank, jaringan perusahaan dan sistem lain yang berisi data sensitif.

Dewasa ini sebagian besar jaringan perusahaan, situs *ecommerce* dan komunitas *online* hanya membutuhkan nama pengguna dan *password* statis untuk *login* dan akses ke dalam data yang bersifat pribadi dan sensitif. Meskipun metode autentikasi ini tidak menyusahkan, *password* statis merupakan bentuk perlindungan yang paling tidak aman terhadap pencurian identitas *online* seperti: *phishing, keyboard logging, serangan man in the middle* dan metode lain.

One time password dapat dihasilkan dalam beberapa cara dan masing-masing memiliki manfaat yang berbeda dari segi keamanan, kenyamanan, biaya dan akurasi. Solusi autentikasi yang kuat mengatasi keterbatasan *password* statis dengan memasukkan langkah keamanan tambahan. *One time password* yang bersifat sementara melindungi akses jaringan dan identitas digital end-user. *One time password* menambahkan tingkat tambahan perlindungan dan membuatnya sangat sulit bagi penipu untuk mengakses informasi yang tidak sah, jaringan atau rekening *online*.[GEM-06]

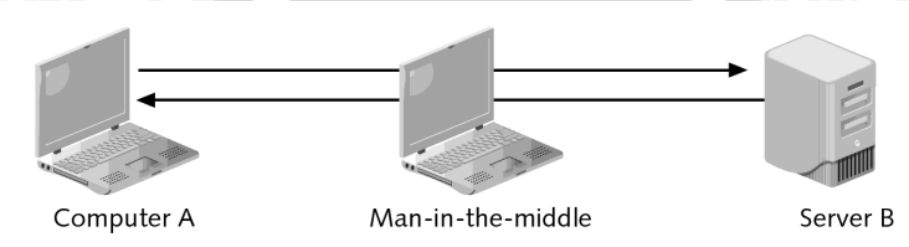
2.6 *Ecommerce*

Electronic commerce (ecommerce) adalah suatu penjualan secara elektronis, yang bisa dilakukan dari jarak jauh (teknologi marketing) yang digunakan di luar toko. Untuk tempat yang jauh sekalipun tetap dilakukan perdagangan dengan memanfaatkan *ecommerce*. Perubahan cara dan bentuk perdagangan telah mengubah, menggeser dan menaklukkan cara bisnis global yang tidak mengenal jarak dan waktu. Kegiatan yang dilakukan juga menjadi tidak banyak lagi diwakili oleh tenaga manusia di saat terjadi peningkatan keterpaduan telekomunikasi dan komputasi secara integral. [TAZ-10]

Pembeli yang akan berbelanja di toko *online* dapat menggunakan fasilitas *shopping cart. Shopping cart* adalah sebuah *software* di situs *web* yang mengijinkan pelanggan untuk melihat toko yang anda buka kemudian memilih item barang untuk diletakkan dalam kereta dorong yang kemudian membelinya saat melakukan *check out.* Konsep *shopping cart* ini meniru kereta belanja yang biasanya digunakan orang untuk berbelanja di pasar swalayan. *Shopping cart* biasanya berupa formulir dalam web.

2.7 *Man In The Middle*

Man in the middle dapat dilakukan pada sebuah jaringan ketika dua komputer saling berkomunikasi. Jenis serangan ini membuat seolah-olah hanya dua komputer yang terlibat saat sedang mengirim dan menerima data, sedangkan sebenarnya diantara mereka terdapat sebuah komputer yang dapat melihat lalu lintas data yang sedang dikirim.



Gambar 2.1 Serangan Man in The Middle
Sumber: [CDM-11]

Serangan *man in the middle* bisa berbentuk pasif atau aktif. Untuk penelitian ini, peneliti menggunakan *man in the middle* berjenis pasif sebagai langkah pengujian keamanan. Pada serangan pasif *man in the middle*, penyerang menangkap data yang terdeteksi.[CDM-11]

2.8 *Brute Force*

Sebuah serangan *brute force* adalah salah satu serangan yang melibatkan kode atau *password* dengan mencoba semua kombinasi yang mungkin sampai yang benar ditemukan. Ini bukan sesuatu yang mudah dilakukan dan mungkin membutuhkan waktu yang sangat lama, tergantung pada jenis enkripsi yang digunakan dan tingkat keamanan pada sistem tertentu. Hal ini dianggap sebagai jalan terakhir setelah penyerang tidak dapat mengakses sistem melalui cara lain . Tergantung pada jumlah atau kunci yang digunakan pada data yang terenkripsi, serangan *brute force* mungkin atau mungkin tidak dipertimbangkan. Saat sebuah kombinasi *password* mempunyai panjang kunci yang signifikan, seorang penyerang dapat memutuskan untuk mencoba memecahkan kode. Probabilitas untuk menemukan kombinasi yang tepat dalam waktu singkat menurun dengan setiap kunci tambahan yang digunakan. Setiap sistem dinilai berdasarkan seberapa mudah atau sulitnya untuk melancarkan serangan *brute force* terhadap sistem tersebut. Sebuah sistem yang bagus tidak akan mudah terganggu oleh serangan ini

dan dengan demikian sistem tersebut akan bernilai tinggi. Dalam teori serangan *brute force* selalu dicapai Namun, ada beberapa yang benar-benar dapat diserang dan akan membutuhkan milyaran tahun untuk menyelesaikan .

Serangan *brute force* dapat sangat efektif ketika berdiri sendiri atau bersama dengan serangan jenis lain. Pada dasarnya serangan *brute force* berhasil berdasarkan dari keahlian penyerang atau kelemahan dari sistem yang diserang. Salah satu cara serangan *brute force* dapat berhasil adalah dengan menggabungkan *dictionary attack* . Penggunaan kata-kata konvensional memudahkan pengguna individu untuk mengingat kode akses mereka , meskipun dengan mengorbankan tingkat keamanan yang lebih baik. Serangan *brute force* dengan mengambil keuntungan dari taktik ini dengan menggunakan *dictionary attack* untuk meningkatkan kemungkinan bahwa kode dari sebuah pengguna dapat ditemukan cukup cepat [ROL-13]. Namun pada penelitian kali ini, penulis tidak menggunakan metode *dictionary attack.* Penulis menggunakan metode *brute force* dengan mencoba kemungkinan yang ada dengan *range* karakter tertentu.

2.9 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan disiplin ilmu yang berkaitan dengan semua aspek produksi perangkat lunak dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah sistem digunakan. Dalam definisi rekayasa perangkat lunak terdapat dua frase kunci:

1. Disiplin rekayasa

   Perekayasa membuat suatu alat bekerja. Mereka menerapkan teori, metode, dan alat-alat dimana dapat digunakan secara tepat. Namun, mereka menggunakan secara selektif dan selalu mencoba untuk menemukan solusi dari suatu masalah bahkan ketika tidak terdapat dalam teori maupun metode. Perekayasa juga mengakui bahwa mereka harus bekerja dalam tekanan organisasi dan finansial sehingga merak mencari solusi dalam kondisi tersebut.

2. Semua aspek dari produksi perangkat lunak

   Perangkat lunak tidak hanya peduli dengan proses teknis dari pengembangan perangkat lunak. Hal ini juga mencakup kegiatan seperti

manajemen proyek perangkat lunak dan pengembangan alat, metode, dan teori untuk mendukung produksi perangkat lunak.

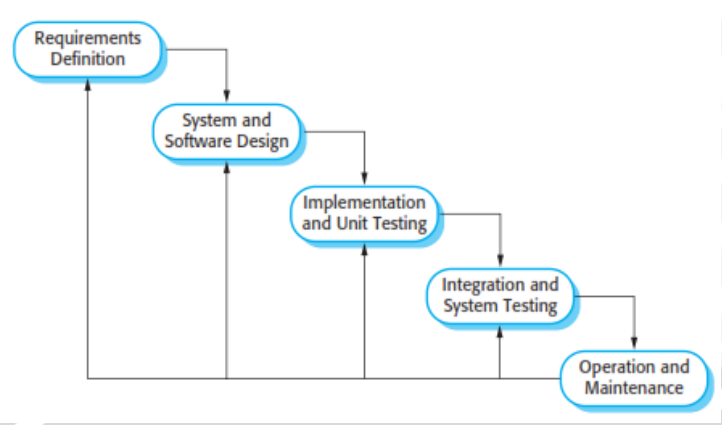
Secara umum, perekayasa perangkat lunak mengadopsi pendekatan yang sistematis dan terorganisir untuk bekerja, karena hal ini cara yang paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Namun, rekayasa adalah segala sesuatu tentang memilih metode yang paling sesuai untuk suatu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan [IAN-07].

2.10 Software Process Model

Model proses perangkat lunak adalah representasi yang sederhana dari proses perangkat lunak. Setiap model proses merupakan proses dari perspektif tertentu, dan dengan demikian hanya menyediakan informasi parsial tentang proses tersebut. Misalnya, model proses aktivitas yang menunjukan kegiatan dan urutan namun tidak menunjukan peran orang yang terlibat dalam aktivitas ini. Sebuah model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat proyek dan aplikasi, metode dan alat-alat yang akan digunakan, dan kontrol *deliverable* yang diperlukan. Beberapa model proses perangkat lunak yang sering digunakan para pengembang perangkat lunak adalah *waterfall model, incremental development,* dan *reuse-oriented software engineering.*

2.10.1 Waterfall Model

Secara umum, perekayasa perangkat lunak memakai pendekatan yang sistematis dan terorganisir terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Namun demikian, rekayasa ini sebenarnya mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan [IAN-07]. Proses pengembangan menggunakan model proses *waterfall* ini terlihat pada Gambar 2.2.



Gambar 2.2 Pemodelan waterfall
Sumber: [IAN-07]

Model proses untuk rekayasa perangkat lunak dipilih sesuai dengan sifat dari proyek dan aplikasi yang akan dibuat. Salah satu dari model proses yang digunakan adalah *waterfall model*. Model proses *waterfall* ini merekomendasikan pendekatan yang sistematis dan terurut (*systematic and sequential approach*) untuk pengembangan perangkat lunak yang dimulai dari analisis kebutuhan (*requirement analysis*), perancangan (*design*), implementasi (*coding*), pengujian (*testing*), dan pemeliharaan (*maintenance*).

2.10.2 Software Reuse

Software Reuse pada dasarnya adalah penggunaan kembali perangkat lunak yang telah ada. Hal ini sering terjadi ketika orang-orang yang bekerja pada proyek mengetahui desain atau kode yang mirip dengan apa yang dibutuhkan. Mereka mencari dan memodifikasi sesuai kebutuhan lalu menggabungkan ke dalam sistem. Pendekatan *reuse* mengandalkan komponen perangkat lunak yang dapat digunakan kembali dan mengintegrasikan kerangka untuk komposisi komponen sistem. [IAN-07]

Tahapan utama dari *reuse* secara langsung mencerminkan dasar pembangunan kegiatan:

1. Analisis komponen

   Mengingat pencarian spesifikasi kebutuhan dilakukan untuk komponen untuk mengimplementasikan spesifikasi tersebut. Biasanya tidak ada

komposisi yang tepat dan komponen yang dapat digunakan hanya menyediakan beberapa fungsi yang diperlukan.

2. Kebutuhan dalam modifikasi

   Selama tahap ini kebutuhan dianalisis menggunakan informasi tentang komponen yang telah ditentukan. Kemudian dimodifikasi untuk mencerminkan komponen yang tersedia. Dimana satu kondisi tidak memungkinkan memodifikasi, kegiatan analisis komponen dapat dilakukan kembali untuk mencari solusi alternatif.

3. Desain sistem dengan *reuse*

   Selama fase ini mendesain *framework* pada sistem atau *framework* yang tersedia akan digunakan kembali. Para desainer memperhitungkan komponen yang digunakan kembali dan mengatur *framework* untuk mengembangkannya.

4. Pengembangan dan integrasi perangkat lunak

   Perangkat lunak yang tidak dapat diperoleh secara eksternal maka akan dikembangkan dan komponen diintegrasikan untuk menciptakan sistem baru.

Pada penelitian ini penulis menggunakan metode *waterfall* yang didalamnya terdapat tahapan *reuse* sehingga tahapan pada penelitian ini adalah: analisis kebutuhan, analisis komponen, perancangan, implementasi, dan pengujian.

2.11 Pengujian Perangkat Lunak

Pengembangan sistem perangkat lunak melibatkan serangkaian kegiatan produksi di mana peluang untuk keteledoran manusia sangat besar. Kesalahan dapat muncul pada awal proses dimana kemungkinan terjadi kekeliruan pada tujuan atau tujuan tidak terspesifikasi secara tepat. Karena ketidakmampuan manusia dalam membuat sesuatu yang sempurna, maka pengembangan perangkat lunak disertai dengan aktivitas penjaminan kualitas. Pengujian perangkat lunak merupakan elemen penting dari jaminan kualitas perangkat lunak dan merepresentasikan tinjauan utama spesifikasi, desain, dan pembuatan kode [PRE-10].

2.11.1 Pengujian Validasi

Pada kulminasi pengujian terintegrasi, perangkat lunak secara lengkap dirakit sebagai suatu paket; kesalahan *interfacing* telah diungkap dan dikoreksi, dan seri akhir dari pengujian perangkat lunak, yaitu pengujian validasi dapat dimulai. Validasi dapat ditentukan dengan berbagai cara, tetapi definisi yang sederhana adalah bahwa validasi berhasil bila perangkat lunak berfungsi dengan cara yang dapat diharapkan secara bertanggung jawab oleh pelanggan. Validasi perangkat lunak dicapai melalui sederetan pengujian *black-box* yang memperlihatkan konformitas dengan persyaratan. Rencana pengujian menguraikan kelas-kelas pengujian yang akan dilakukan, dan prosedur pengujian menentukan *test case* spesifik yang akan digunakan untuk mengungkap kesalahan dalam konformitas dengan persyaratan. Baik rencana dan prosedur didesain untuk memastikan apakah semua persyaratan fungsional dipenuhi; semua persyaratan kinerja dicapai; dokumentasi betul dan direkayasa oleh manusia: dan persyaratan lainnya dipenuhi (transportabilitas, kompatibilitas, pembetulan kesalahan, maintanabilitas)

Black-box testing atau *behavioral testing* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black-box* merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan [PRE-10]

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface.*
3. Kesalahan dalam struktur data atau akses *basis data* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi

2.11.2 User Acceptance Testing

User Acceptance Testing (UAT) adalah salah satu prosedur proyek perangkat lunak akhir dan kritis yang harus terjadi sebelum perangkat lunak baru dikembangkan akan tergulir keluar ke pasar. Selama UAT, pengguna perangkat lunak sebenarnya menguji perangkat lunak untuk memastikan dapat menangani tugas-tugas yang diperlukan dalam skenario dunia nyata dan sesuai dengan spesifikasi [MAX-13].

UAT langsung melibatkan pengguna yang dituju perangkat lunak. UAT dapat diimplementasikan dengan membuat perangkat lunak yang tersedia untuk percobaan *beta* gratis di internet atau melalui tim pengujian yang terdiri dari pengguna perangkat lunak yang sebenarnya

2.11.3 Pengujian Keamanan

Pengujian (*Testing*) keamanan saat proses autentikasi dilakukan dengan dua jenis serangan yaitu *Man In The Middle* dan *Brute Force*. Untuk serangan Man In The Middle digunakan software penyadap *Wireshark*. *Wireshark* adalah sebuah *network packet analyzer* yang mencoba menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut sedetail mungkin [FAG-11].

Sedangkan untuk serangan *brute force*, peneliti menggunakan program *Fireforce* dan program yang dibuat dengan menggunakan CURL. *FireForce* adalah ekstensi *Firefox* yang dirancang untuk melakukan serangan *brute-force* pada *form GET* dan *POST*. *FireForce* dapat menggunakan kamus atau menghasilkan *password* berdasarkan beberapa jenis karakter. *FireForce* dapat digunakan pada *platform* apa saja yang menjalankan *browser web Firefox* [FRF-13]. CURL alias URL Client merupakan sebuah *library* yang berbasis *command line*, dimana dapat digunakan untuk memasukkan parameter ke dalam *web request*. *Libcurl* membuat semua jenis komunikasi antar sever mungkin terjadi dengan berbagai macam cara. Bisa dengan protokol http, https, ftp, gopher, telnet, dict, dan ldap dengan dukungan HTTPS *certificate*, HTTP POST, HTTP PUT, FTP *uploading*, *upload* berbasis HTTP *form*, proxy, cookies, bahkan autentikasi user dan password. Curl adalah *porting* ke PHP sebagai modul opsional dan dapat

berguna untuk mendapatkan informasi pengintaian, atau akses tidak sah ke URL yang ditunjuk. Curl dapat digunakan bersama dengan *script* PHP untuk serangan *brute force*, serangan pengintai, *spoofing*, dan pencurian data. [HAK-13