OTENTIKASI SISTEM DENGAN MENGGUNAKAN *ONE TIME PASSWORD* MEMANFAATKAN *SMARTPHONE ANDROID*

Mohammad Noor Al Azam

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Narotama Surabaya
dosen@noorazam.my.id

Abstrak

Seiring perkembangan Teknologi Informasi yang begitu pesat, dan pertumbuhan infrastruktur di dunia maya, perkembangan situs juga semakin bertambah. Semakin banyak situs yang ada, membuat pengguna internet harus memiliki akun di berbagai situs tersebut. Dengan banyaknya akun, maka semakin banyak pula *password* yang harus dihapalkan untuk melakukan *login*. Hal ini membuat semakin sulit pula untuk mengingat *username* dan *password*. Selain itu, dengan semakin banyaknya *username* dan *password* yang harus diingat, membuat pengguna lebih memilih *password* yang mudah diingatnya. Atau pengguna cenderung menggunakan fasilitas "*keep login*" pada sebuah situs agar tidak selalu memasukkan *password* . Hal ini tentu saja membuat resiko bobolnya akun pengguna tersebut lebih besar. Untuk itu penulis berinovasi untuk membuat sebuah aplikasi dengan memanfaatkan *smartphone android* yang tersinkronisasi dengan server, dan dapat memberi solusi pada masalah tersebut. Pengguna tidak lagi perlu menghafal masing-masing *password* untuk masing-masing situs. Pengguna cukup memilih situs yang akan dimasukinya pada sebuah aplikasi *Android*, dan aplikasi itu akan melakukan sinkronisasi dengan server dan memberikan *one time password* (OTP) yang hanya berlaku selama 5 menit.

Kata Kunci : *One Time Password, Synchronize Password, Android*

1. Pendahuluan

Perkembangan teknologi informasi dewasa ini membuat keamanan merupakan suatu hal yang sangat penting. Baik itu keamanan fisik, keamanan data maupun keamanan aplikasi. Salah satu cara melakukan pengamanan adalah menggunakan otentikasi *username* dan *password*. Berdasarkan penjelasan diatas penulis mempunyai berinovasi mengembangkan sebuah sistem otentikasi yang lebih aman dengan menggunakan *smartphone Android* yang saat ini banyak digunakan.

Dengan menggunakan aplikasi ini, maka pengguna tidak lagi perlu menghapalkan masing-masing pasangan *username* dan *password* untuk semua situs yang memerlukan otentikasi. Pengguna cukup melakukan pemilihan situs yang akan dimasukinya dan nanti aplikasi akan memberikan informasi *username* apa yang digunakan dan *password* apa yang berlaku saat itu. *Password* ini berlaku hanya satu kali *login* dan hanya berlaku selama beberapa menit saja.

2. Tinjauan Pustaka

2.1. Android

Android adalah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi dan aplikasi. Seiring perkembangannya android berubah menjadi platform yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembang utama dibelakangnya yaitu google. Sebagai generasi baru *platform mobile*, android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi dan memberikan kesempatan kepada pengembang secara leluasa untuk menciptakan aplikasi sesuai dengan yang diharapkan.

2.2. PHP

PHP atau yang memiliki kepanjangan PHP Hypertext Preprocessor merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML

digunakan sebagai pembangun atau pondasi dari kerangka layout web, sedangkan PHP difungsikan sebagai prosesnya.

2.3. MySQL

SQL merupakan kependekan Structured Query language . SQL digunakan untuk berkomunikasi dengan sebuah database. SQL adalah bahasa yang meliputi perintah-perintah untuk menyimpan, menerima, memelihara, dan mengatur aksesakses ke basis data serta digunakan untuk memanipulasi dan menampilkan data dari database.

2.4. Otentikasi

Otentikasi, berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri, untuk validasi user pada saat memasuki sistem. Dalam hal ini autentikasi merupakan sebuah proses identifikasi yang dilakukan oleh pihak yang satu terhadap pihak yang lain ataupun sebaliknya dengan melakukan berbagai proses identifikasi untuk memastikan keaslian dari informasi yang diterima.

3. Metodologi Penelitian

3.1. Analisa Sistem

Sistem OTP ini memiliki 2 sub-sistem yang berdiri sendiri. Satu sistem ada di sisi server (*server-side*) dan satu sistem lagi adalah aplikasi yang harus diinstall di dalam *smartphone Android* (*mobile-side*). Saat melakukan instalasi aplikasi *mobile-side*, selain meminta *username* pengguna untuk situs yang dituju, aplikasi juga membuat sebuah angka desimal acak, antara 1.000.000 sampai dengan 9.999.999. Angka ini sengaja dibuat 6 digit untuk mengurangi kemungkinan kesamaan angka yang muncul.

Setelah mendapatkan angka desimal yang acak, *mobile-side* akan menginformasikan *username*, nama situs dan angka acak ini ke *server-side* sebagai referensi untuk pembuatan OTP nantinya. *Password* yang dibuat oleh sistem akan berdasarkan pada *hash* kesatuan informasi *username*, angka acak yang sudah dibuat sebelumnya, jam dan menit saat dibuat OTP. Dengan adanya informasi pewaktu dalam *hash* ini, maka *hash* yang dihasilkan akan selalu berbeda setiap saat.

3.2. Perancangan Desain & Sistem

Dua buah sub-sistem yang dibuat akan menggunakan referensi informasi yang sama. *Username* dan angka acak sudah dimasukkan sebelumnya oleh pengguna. Sementara nilai pewaktu yang ditambahkan di dalamnya, akan dilihat berdasarkan waktu penggunaan OTP. Oleh karenanya, *server-side* dan *mobile-side* harus memiliki informasi yang sama dengan pewaktu dunia. Hal ini sudah bisa dilakukan karena *Android* dan server bisa menggunakan *time server* dunia yang ada.

Pada saat otentikasi dilakukan, *mobile side* akan membuat sebuah *hash* berdasarkan informasi-informasi yang sudah dimilikinya tersebut. *Hash* itu akan menjadi sebuah *password* bagi *username* itu untuk masuk ke sebuah situs. Karena *hash* memiliki informasi pewaktu berupa jam dan menit, maka *server-side* harus melakukan hal yang sama. Namun untuk mengantisipasi *delay* yang terjadi, *server-side* harus membandingkan pewaktu 3 menit sebelum dan 3 menit sesudah pewaktu server saat itu. Pembandingan 3 menit ini berarti umur OTP tersebut maksimal adalah 6 menit setelah dibuat oleh *mobile-side*.

4. Hasil & Pembahasan

Mobile-side membutuhkan *SQLite* sebagai tempat menyimpan informasi *username* dan angka acak yang sudah dibuat pada saat instalasi. Sementara aplikasi *server-side* membutuhkan Apache atau NGINX sebagai web server yang sudah dilengkali dengan PHP *pre-processor* dan MySQL sebagai tempat penyimpanan informasi *username* dan angka acak yang didaftarkan oleh pengguna sebelumnya.

Pada saat permintaan OTP, pengguna harus membuka aplikasi *mobile-side*. Aplikasi ini kemudian akan melakukan *hash MD5* gabungan *username*, angka acak, jam sistem saat itu dan menit sistem saat itu (“<username><angkaacak><jam><menit>”). Hasil dari *hashing* ini akan diinformasikan melalui layar kepada pengguna sebagai *password*-nya.

Sementara itu, pada sisi *server-side* untuk membandingkan *password* yang dimasukkan oleh pengguna menggunakan metode yang sama. *Server-side* harus melakukan *hash MD5* gabungan informasi *username* dan nilai acak yang ada di dalam MySQL, dan ditambah dengan jam dan menit saat permintaan otentikasi itu dilakukan. Hasil dari *hashing* ini dicocokkan dengan yang sudah dimasukkan oleh pengguna ke dalam sistem.

Untuk menghindari *delay* yang ada, menit yang digunakan dalam *hashing* oleh *server-side* ini dilakukan dimulai dari -3 dan +3 pewaktu saat *hashing* dilakukan.

4.1. Perancangan Sistem

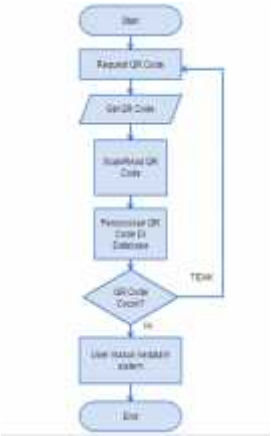
Perancangan sistem ini dibagi dalam dua tahap. Yaitu tahap registrasi yang proses kerjanya seperti pada gambar 1.

Gambar 4.1. Proses Registrasi.

Pada tahap registrasi, user diminta untuk memberikan input *username* dan *mobile-side* membuatkan nilai acak yang hendak didaftarkan. Terdapat prosedur standar pemeriksaan pada tahap registrasi ini. Apabila pengguna tidak memasukkan input pada *textbox* atau menggunakan *username* yang tidak sesuai dengan peraturan yang ada, kemudian menekan tombol registrasi. Sistem akan memberikan peringatan kepada pengguna untuk melengkapi pendaftaran. Setelah penginputan *username* selesai sistem akan melakukan *generate* nilai acak secara otomatis dan menyimpan hasil *generate* tersebut ke dalam database.

4.1.1. Proses Otentikasi

Setelah proses registrasi selesai user dapat melakukan *request* OTP yang nantinya akan digunakan pada saat *login* di situs yang telah terdapat data pengguna tersebut, yang alurnya dapat dilihat pada gambar 2.



Gambar 4.2. Proses Otentikasi.

Setelah pengguna mendapatkan hasil *hashing* yang dilakukan oleh *mobile-side*, nilai itu harus dimasukkan sebagai *password*-nya. Dan karena *hashing* dilakukan dengan informasi pewaktu, maka nilai *hash* akan selalu berbeda setiap waktu.

Selain itu, informasi pewaktu ini juga membuat umur *password* OTP dapat disesuaikan dengan kebutuhan sistem. Caranya dengan melakukan pengecekan dalam periode tertentu. Jika melebihi periode itu, maka hasil *hashing* itu akan tidak lagi berlaku bagi *server-side* dan pasti ditolak oleh sistem.

4.1.2. Tampilan Antarmuka

Tampilan antarmuka server dibuat seperti gambar 3.

Gambar 4.3. Tampilan halaman login.

Setelah memasukkan *username* dan OTP yang ada, maka pengguna akan diijinkan masuk dan proses identifikasi akan menggunakan *session* dan *cookies* seperti situs-situs biasa.

4.1.3. Tampilan Antarmuka Perangkat

Tampilan antarmuka pada perangkat android didesain sangat sederhana. Dimana pada saat pertama kali user membuka aplikasi, terdapat pilihan tombol *register* dan *exit*. Untuk memperoleh OTP. Pengguna harus melakukan registrasi terlebih dahulu dengan mengisi *form* yang tersedia berupa *username* dan *password* yang diinginkan.

Setelah itu, untuk mendapatkan OTP, pengguna harus memilik *generate* dan hasil dari dari aplikasi ini akan menjadi OTP yang harus dimasukkan ke dalam *server-side*.

Gambar 4.4. Tampilan pada perangkat android.

Daftar Pustaka

Agus Saputra. (2011). *Trik Dan Solusi Jitu Pemrograman PHP.* Jakarta: Elex Media Komputindo.

Gargenta. (2011). *Learning Android*. California: O'Rilley Media.

http://www.hjp.at/doc/rfc/rfc6238.html

Meier. (2010). *Professional Android 2 Application Development*. California: Wiley.

Nababan. (2011). *Studi Perbandingan Antara Metode Probabilistic Encryption dengan Metode Rivest Shamir Adleman.* Medan: Universitas Sumatra Utara.

Rosari, R.W. (2008). *PHP Dan MYSQL Untuk Pemula*. Yogyakarta: ANDI.

Rouillard. (2008). *Multimodality in Mobile Computing and Mobile Devices. Hershey, New York: IGI Global.*

Subpratatsavee, Puchong; Kuacharoen, Pramote. (2015). *Internet Banking Transaction Authentication Using Mobile One-Time Password and QR Code*. California: American Scientific Publishers